THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Douglas Gregg Davey

TITLE OF THESIS A Computational Model of Language Acquisition

DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science

YEAR THIS DEGREE GRANTED 1979

THE UNIVERSITY OF ALBERTA

A COMPUTATIONAL MODEL OF LANGUAGE ACQUISITION

BY

DOUGLAS GREGG DAVEY

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

FALL, 1979

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled A COMPUTATIONAL MODEL OF LANGUAGE ACQUISITION submitted by Douglas Gregg Davey in partial fulfilment of the requirements for the degree of Master of Science in Computing Science.

ABSTRACT

Interest in the computational modelling of natural language acquisition has grown in both the fields of Computer Science and Psychology, yet for a variety of reasons, such modelling remains in its infancy.

Several of the more recent models of language acquisition are reviewed and an indication of where the scope of such models could be broadened is given.

A model incorporating several sub-tasks of language acquisition including grammar, concept and some vocabulary acquisition is then presented.

Several experiments are described, which serve to illustrate the effectiveness of the current model as well as its individual components.

Finally, a number of the model's shortcomings are documented and possible resolutions to these difficulties as well as an indication of where further works remains to be done, is given.

ACKNOWLEDGEMENTS

I would like to thank Kelly Wilson, for his moral support which lifted me from the doldrums of my unofficial sabbatical leave, and Jeff Sampson for his time and financial support which enabled me to spend as much time as I did on this research.

Also, somewhat reluctantly, I wish to acknowledge the many fine friends who did their best to see that this thesis took as long as possible to complete.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Artificial Intelligence and Language Acquisition

The research reported herein is primarily concerned with the Artificial Intelligence approach to the modelling of language acquisition. Definitions of what constitutes Artificial Intelligence vary somewhat as can be seen from the following quotes. "Artificial Intelligence is the ability of machines to do things that people would say require intelligence," (Jackson, 1975) or "Artificial Intelligence is concerned with the creation of computer programs capable of performing tasks normally considered (by people) to require intelligence," (Charniak, 1976). These definitions are problematic in that they are not precise and are subject to varying interpretations as the knowledge of intelligence expands. A more pragmatic way to view Artificial Intelligence is to examine its goals; "Artificial Intelligence is the study of ideas which enable computers to do the things that make people seem intelligent," (Winston, 1977) or "The central goals of Artificial Intelligence are to make computers more useful and to understand the principles which make intelligence possible," (Winston, 1977). In accord with these goals is the study of those aspects of intelligence which are necessary for the use of language. The growing interest in language, whether in the

understanding or representation of it, can readily be seen by examining the contents of the past few Proceedings of the International Joint Conferences on Artificial Intelligence. The reason for the focus on language is that it is the means through which models of intelligence can understand and interact with the real world.

Within the Artificial Intelligence paradigm one of the least understood aspects of language is the method by which it may be acquired. This is unfortunate in that there are many benefits to be realized from an understanding of the acquisition process. One of the benefits is that the details of how language is learned may have correspondences with how other knowledge is acquired; it is doubtful that the two processes are completely separate. Another benefit is that such an understanding of language could provide more flexibiltiy for intelligence models which encounter novel input, or unfamiliar language constructs. In this case it is just not feasible to prepare a model to cope with all the vagaries of the real world. More importantly however, an understanding of how language is acquired could lead to a more efficient analysis of the knowledge underlying language and to the programming of such knowledge. At present this knowledge is hand programmed and, because of the inherent detail, extremely tedious.

To illustrate this final point, consider Schank and Abelson's (1977) story understanding model. The model

incorporates specific world knowledge similar to that which people use to interpret and participate in events they have been through many times. This knowledge permits relatively little processing of and wondering about frequently experienced events. The knowledge about a particular event is organized into a set of ordered scenes which in turn consist of groupings of causal chains. This information, together with a list of relevent objects, roles, prior conditions and possible results, comprises a "script" for the handling of a common event. It is claimed that such information is necessary to understand a story as simple as,

> John went to a restaurant
> He ordered chicken
> He left a large tip

In this example the script is required to infer that "John" was served and that he was quite satisfied with his meal.

It is due to the lack of automation in the acquisition of such knowledge that Artificial Intelligence models are only able to deal with small segments of the real world. Hopefully, a better understanding of how language is acquired will aid in this endeavour.

1.2 Psychology and Language Acquisition

Interest in computational models of language can also be found in Psychology. The psychologist is interested in language since it is the main medium through which he can

study human knowledge systems. The intent of such models is to simulate as closely as possible the human learning behavior. Often models outside of Psychology explicitly express a similar intent, though probably as frequently they do so implicitly. For the most part, no model excludes some characteristics of human learning entirely; the failure of purely heuristic models, particularly in regards to language translation of the early 1960s, is reasonably well known.

There is some difficulty, however, in dealing with the empirical knowledge of how children acquire language. The problem with this information is that it consists of a vast amount of scattered observations recorded in a variety of ways, and as yet, there is no complete and cohesive theory which can tie this descriptive information together. Also, in these observations the samples used are often small and/or studied for only brief periods of time. Perhaps the greatest defect in the knowledge of acquisition is that what is known is in the main descriptive, rather than explanatory. More aptly put, the availabe data is of the form, "this is what occurred," versus, "this is how it occurred". Clearly, this is a particularly significant defect for any computational modelling endeavour and so any claim that a model simulates human behavior should be tempered somewhat to claim that the model exhibits certain characteristics of human behavior.

Though the goals of acquisition models in Psychology

and Artificial Intelligence are slightly different they do share many common characteristics. Actually, Schank and Abelson (1977) claim that the orientations of Psychology and Artificial Intelligence interesect when, "the psychologist and computer scientist agree that the best way to approach the problem of building an intelligent machine is to emulate the human conceptual mechanisms that deal with language." While the aim of the current research was not to emulate the human conceptual mechanisms underlying language, for reasons discussed above, it was also not intended to ignore the results of experimental psychology. It was found that such results were one source of ideas for the theoretical beginnings upon which the research could be based. The ideas are more fully documented in the description of the model presented in Chapter 3.

1.3 Background for Language Acquisition

There is universal agreement that a certain level of cognitive knowledge be present before language can begin to develop, a point acknowledged in both Psychological and Artificial Intelligence paradigms. Unfortunately, the knowledge assumed by the models remains relatively static and does not develop with the language being acquired. No doubt this is a result of the lack of a general model of cognitive development with sufficient detail for computational purposes. Considering the state of most current research this may soon become an inhibiting factor.

There is some controversy over the nature of this assumed cognitive knowledge. On the one hand, since language is largely confined to man, it is thought that he must have some innate cognitive ability specific to the acquisition of language, a position held by Chomsky (1965). On the other hand, it is claimed that the development of cognitive knowledge is a result of a person's interaction with his environment. The general consensus, Chomskians aside, is that whatever innate knowledge humans (models) possess, it is best thought of as being a general cognitive ability rather than something specific to language (Gardner & Gardner, 1975; Sinclair-de Zwart, 1973). Since for the most part, models of language acquisition do not entertain the question of cognitive development, no further discussion of these points will take place.

The cognitive knowledge that is assumed manifests itself in a variety of forms. Typically, all the concepts corresponding to objects and actions which can be experienced are known, but have not yet been associated with the surface strings of a language. In some cases conceptual relations (knowledge of conceptual classes) is specified and in certain models grammatical knowledge on the ordering of conceptual classes is also provided. The effect of these assumptions can be significant and perhaps a little subtle. It is this initial core of knowledge that will eventually determine what a model can learn, the manner and flexibility

with which it can be done, and even the rate at which learning can take place. In a simulation sense, the selection of the level and amount of cognitive development can be particularly significant. The selection of a too comprehensive or perhaps a too complete core of knowledge could invalidate the results of a model which supposedly explains human behavior. It is likely that the steps taken in the acquisition process would probably be most unlike those actually taken. In the Artificial Intelligence paradigm these assumptions regarding cognitive development may not be as important, but since they affect the overall acquisition process, they should be considered in the study of any language acquisition model.

1.4 Nature of Language Acquisition Models

The actual nature of the acquisition process tends to be more inductive than deductive in that the analysis involved often works with faulty or incomplete knowledge. In contrast, language comprehension models, with an acquisition component, are significantly more deductive as is exhibited in Granger's (1977) program FOUL-UP. The program is activated whenever an unknown word appears in its input and through the use of internal parsing expectations and given world knowledge, the program is able to deduce a context specific definition.

It is universally the case that a sentence of natural

language plus the corresponding environmental context provides the basic unit of input supplied to a language acquisition model. Occasionally several instances of the environmental context will be provided so as to allow for the possible procedural activities which may take place. Additionally, "attention" indicators may be given to a model to allow it to focus on a smaller set of data and in some cases, expert feedback is used to guide a model along the "correct" lines of acquisition. Once given this basic unit of input the model can initiate its acquisition processes.

For the most part, the acquisition of word meanings and rudimentary grammar have been the two main tasks that acquisition models have concentrated on. Most of the significant models currently consider only vocabulary, or only grammar acquisition. While the current aim of all acquisition models to date is to gain some mastery in child-like language, this may appear to be a somewhat limited goal when compared with state-of-the-art language comprehension models. However, despite dealing with the same phenomena, acquisition models attempt to deal with problems that either do not arise or are insignificant as far as comprehension models are concerned. A serious deficiency however, is that input to acquisition models is still restricted to simple declarative sentences with no consideration being made for the handling of paragraph length text and the corresponding problem of dealing with connected discourse.

Another distinguishing characteristic of language acquisition models is the amount of freedom of operation they possess. The ideal model would have a curiousity or question asking component such that no overt teaching would be necessary. That is, the model would not require to be "led by the hand" through every step of the learning process. It should be able to encounter unrestricted natural language, make errors yet recover, and operate independently from an expert teacher. Of course no model as yet approaches this ideal, even remotely, but there is an appreciable difference among current models in this regard.

With the aims of acquisition models focused on learning child-like language, the transition to full adult-like language is a task which has not been considred in any detail. The ultimate goal of any acquisition model is to reach at least the language manipulating skills of a language comprehension model, though this may likely only occur at some distant date. The intent of the current research is to provide an acquisition model which exhibits the abilities to handle some vocabulary, grammar and conceptual acquisition in as unrestricted a manner as is feasible.

1.5 Computational Modelling

The necessity of a computational model for the computer scientist is obvious, yet for psychologists or others who

deal with complicated models, the benefits of a computer are just as as important. As Reeker(1975) points out, "Models of very complex systems are likely to be so complex that simulation provides the only feasible method for determining their behavior." Also, "The computational modelling process forces a degree of explicitness that is often absent in discursive expositions of theories, ..." There are a number of inherent dangers in computer modelling which Reeker also notes. "It is easy to get carried away when the execution of a complex computer model produces 'interesting' behavior." One must remember that, "... there is no reason to assume that its (the computer) organization in any way reflects the organization of any naturally-occurring system." This argument also applies to whatever programming language that is used. Hence one must be "careful to separate legitimate theoretical constructs from the the constructs required by the modelling medium."

Chapter 2

RELATED RESEARCH

In this chapter a number of the more recent computational models of language acquisition are examined. The models have been organized into the previously mentioned categories of vocabulary acquisition, structural acquisition and comprehensive acquisition. A pictorial representation of the information sources and tasks performed by these

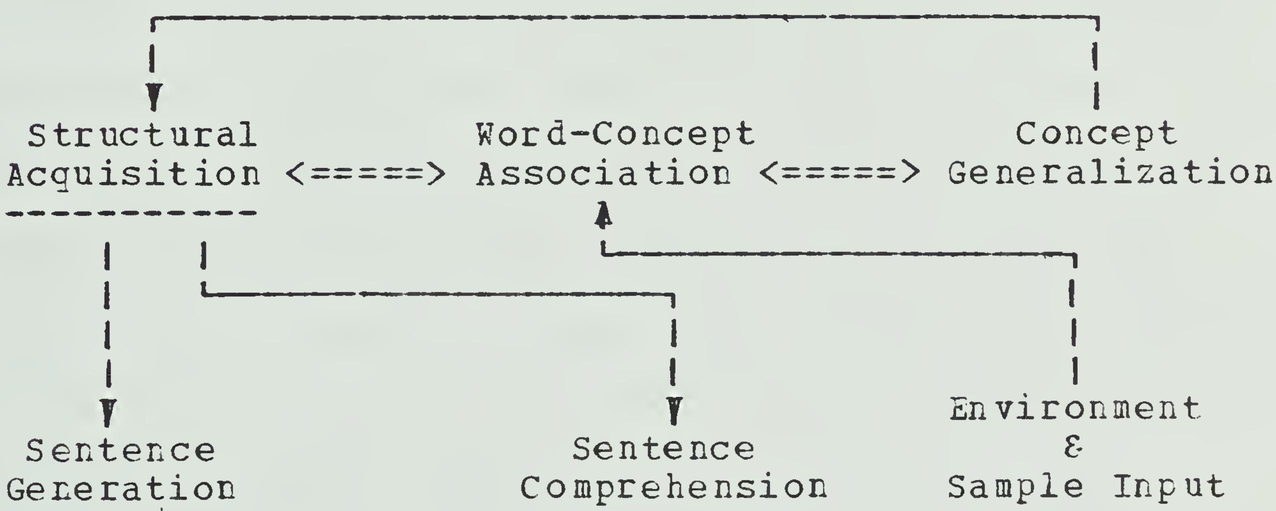


Figure 2.1

models can be found in figure 2.1.

In general, a vocabulary acquisition model attempts to associate a basic conceptual unit to a surface word of a language through the use of sample input and some environmental knowledge. The conceptual units are assumed

to be known by the model and as yet there has been little effort expended in modelling how these concepts are acquired. A structural acquisiton model adds a level of complexity over vocabulary acquisition by endeavouring to also acquire the grammar of a language. Models of this type may or may not include vocabulary acquisition as a component, but clearly a significant number of words must be known by the model before grammar can be acquired. Because of this, some structural acquisition models forego vocabulary acquisition and simply assume the necessary knowledge. As with vocabulary acquisition, sample input and environmental knowledge are also required for the acquisition process. The final category used is that of comprehensive acquisition. Here a further level of complexity is added by attempting to acquire some of the conceptual knowledge that was assumed by the above two categories. A complete comprehensive model would encompass all of figure 2.1 and hence should be in a position to provide one potential explanation of the subtasks of language acquisition and their possible interactions.

2.1 Vocabulary Acquisition

This section will look at three models of vocabulary acquisition. The work of McMaster(1975,1976) is related to that of King(1976) and together they provide models of vocabulary acquisition. Salveter(1976), like King, concentrates on the acquisition of verbal concepts.

2.1.1 A Vocabulary Acquisition System

McMaster has proposed a Comprehensive Language Acquisition Program, though only the subset, a Vocabulary Acquisition System has been implemented. This subset (VAS) engages the problem of acquiring initial word-concept pairs. The program operates in a simple blocks world environment and allows for unrestricted or noisy linguistic input. This blocks world being similar to that of Winograd (1972).

World knowledge is semantically represented by one- and two-place predicates which describe object attributes, class membership of objects and other predicates, and the relative physical position of objects. There are three classes of predicates in VAS,

1. one-place attributive predicate
   (#MANIP x)--"x is manipulative"
2. two-place attribute predicate
   (#IS x y)--"x is y"
3. relational predicate
   (#SUPPORT x y)--"x supports y"

Since VAS is not concerned with grammatical learning there is no differentiation made as to a word's syntactic class.

Input to the program consists of single sentences with corresponding semantic referents or "focal regions". The focal region, which is manually provided to the model, is a list of the internal names of the objects which are probably

described by the given sentence. The acquisition process consists of relating the words of the sentence with the objects contained in the focal region. As sentences are processed, an association count is maintained to reflect the frequency that a word and concept have been associated and additionally, each time a word or concept is used, a usage count is updated to indicate their frequency of reference. These counts are ultimately used to help determine which particular concept most likely corresponds to a given word. Because of time and resource limitations McMaster selected a set of words to be used as candidates for learning.

As mentioned above, the initial semantic referent is only a list of objects possibly described by the input sentence. Hence, prior to the forming of associations, the semantic referent is expanded to contain all "relevent" concepts from the world knowledge. This expansion is accomplished by examining each concept "c" in the referent in conjunction with processing each of the three classes of predicates (p) as follows:

1. For each class 1 predicate p in which c occurs, p is added to the referent.
2. For class 2 predicate p in which c appears as a first argument, each concept in the second argument is added to the referent.
3. For each class 3 predicate p in which c occurs as the first argument and one of the concepts in the second argument also appears in the referent, p is added to the referent.

The first step insures that all the attributes of the

objects in the initial referent are included in the expanded referent. Step two provides the class membership predicates of the objects and other predicates in the referent while step three adds the relational information about the relevant objects. If necessary, the steps of this procedure are applied recursively until no further action can take place.

After this expansion, the word association process can begin. There are four steps involved:

1. All words in the sentence have their usage count incremented.

2. All concepts associated with a word and which are in the focus, have their usage count incremented.

3. If necessary, new concepts are added with a usage count initialized to 1.

4. All concepts in the focus have their usage count incremented.

There are four identifiable categories of associations which can arise from the above steps; not all of which lead to meaningful associations. One such case occurs when a word appears frequently in the input sentences and an associated concept also occurs in a large number of the referents which results in a large association weight being built between the word and concept. If this criterion were used to choose the probable meaning of the word, it is most likely that the program will be in error. Using a high association value to assign a high frequency concept as the meaning of a word, would result in many words being assigned

the most frequently occuring concept as their meaning. Similarly, when a word occurs infrequently but an associated concept is present in a large number of referents, it is unlikely that the concept should be assigned as the word's meaning. It is only the high frequency of the concept which causes a high association weight. If a word occurs frequently, but an associated concept appears in a small number of foci, this is once again a situation similar to the above. It is unlikely that the concept corresponds to the meaning of the word and besides, other concepts probably have a higher association weight. The most promising case arises when both word and concept appear infrequently since whenever the word occurs so does the concept.

The above reasoning was implemented in the model through the use of an evaluation function,

$$F(w,c) = u(c,w)[2 - m(u(c))/u(w)]$$

Variables in the function include the concept usage count "$u(c)$", the word usage count "$u(w)$" and the association count "$u(c,w)$". The constant "m" was determined experimentally and was found to give the highest score of correctly learned words with a value of .21. The significance of this function is that words with high association values will be ranked high. Also concepts with very high or very low frequency will be ranked low if the ratio $u(c,w)/u(c)$ is constant. The highest ranked concepts will be those having usage counts somewhere in the middle of

the interval (0,t) where t is the maximum possible usage count. Finally, if both the concept and word have a high usage count, then the concept will get a higher ranking than it would if the word usage were low.

After the application of the above function, those concepts with the highest rating were chosen to be a given word's probable meaning. The evaluation of the results of VAS are somewhat subjective since the program does not display any overt behavior. Using noisy input and 219 sets of data, VAS correctly learned 9 of 16 words; in a more controlled test, 18 of 24 words were acquired.

There are three situations where VAS is unable to learn the correct meaning of a word. One is where two concepts always appear together. Chance ordering of concepts and round-off errors then determine which concept is chosen. Also if a sentence contains a word whose meaning does not occur in the referent, the program cannot help but fail. Finally, it may happen that the concept corresponding to a meaning of a word has such a high usage count, that the evaluation function chooses another concept instead.

It is significant that VAS achieved some success in acquiring word meanings despite the noisy input provided and despite the absence of any expert feedback to guide the learning process. However it is not clear just how important the constant "m" is in the evaluation function. Whether different values of "m" with different sets of input

will provide an improvement or degradation in the results is unknown. As a simple extension to VAS, it would have been interesting to see how the program would perform after it decided it had acquired a few words. By doing this, perhaps one would gain a better understanding of how knowedge of one word can influence the acquisition of others.

2.1.2 A Verbal Acquisition System

King's(1976) Verbal Acquisition Model (VAM) is closely related to Mcmaster's(1975) Vocabulary Acquisition System. The major difference is that VAS is able to deal with conceptual actions and the corresponding verbs. Another significant difference between the two models is King's use of a case grammar semantic representation rather than the predicate calculus of McMaster. Such a representation can strongly influence the nature of the learning process. If the appropriate verbal structure can be identified, then a wealth of information is available to aid in the selection of possible candidate concepts for the words unknown to the model. Additional differences from McMaster's program include the ability to construct semantic referents from spatial co-ordinates and the allowance for movement which enables the use of a richer vocabulary. The program is also able to alter semantic referents through the manipulation of objects and it is possible for the program to create its own lexicon instead of having it predefined. Like VAS, VAM operates within a blocks world and can see and perform

actions as well as having its attention directed towards a particular location. The usual concept and word usage counts and association counts are also kept.

The semantic representation used is based on Schanks's(1973, 1975) Conceptual Dependency Theory but is somewhat simpler in that it is "tailored" to the blocks world environment. Attributes associated with objects include existence, size, shape, location, color and possibly containment. The primitive actions considered are the physical ones of PROPEL, MOVE and GRASP, and the global PTRANS. The three cases which can be associated with an action are those of actor (VAM itself), instrument and direction. PTRANS has been extended to include rotations about the x-, y- and z-axes or DROP(x), TURN(y) and TWIST(z) respectively. RELOC is used to indicate that an object maintains its original definition in regards to rotations.

These actions, in slightly more detail are described below.

<u>PTRANS</u>--cause an object to change states

cases: agent, object, direction
result: new location or different facing

<u>PROPEL</u>--apply a force to an object, ungrasping in the process

cases: agent, object, direction
instrument: ungrasp of object by agent
result: contents of arm are empty

<u>GRASP</u>--to grasp or let go of an object

cases: agent, object

result: contents of arm is object or empty

MOVE--to relocate the arm

cases: agent, object, direction
result: arm and contents in a new position

In building a semantic referent the necessary primitive actions, which are in fact program names, must be included. Each primitive action can have only a single object and agent and these are also added to the referent. Additionally, since there may exist relations between objects, these too must be placed in the referent. The possible relations considered were BESIDE, FRONT-OF, SUPPORTS, BIGGER and INBOX. Finally, the necessary state changes on an object's attribute values are noted and the referent is expanded as in VAS.

One of the programs in VAM is WATCH, which notices an effect on the environment. WATCH implies the primitive action PTRANS. Hence if there is an object k, with attributes a(k) and relations r(k) with other objects o(k) the referent of PTRANS would be

WATCH PTRANS(TWIST, DROP, TURN or RELOC) k a(k) r(k) o(k) a(o(k))

The program will then move k to its new position before expanding the referent.

If VAM moves its arm the referent would be

MOVEARM VAM MOVE arm a(arm)
[VAM PTRANS k a(k) r(k) o(k) a(o(k))]

where [...] means conditional inclusion.

For VAM to get an object k, requires the GRASPing of k with the possible un-GRASPing of a previous 'k and/or moving to k. In a similar fashion VAM can let go of an object or transfer or turn an object by MOVEing and PTRANSing a grasped object.

Two sets of input were presented to the program; one which included semi-natural conversation and another designed to obtain the program's optimal performance. As in VAS the evaluation of results is somewhat subjective. The results were generally good and for VAM's specialty of learning verbs there were few verbs for which an English equivalent was not attached.

The significance of VAM is the recognition of the need for a strong semantic representation. The use of this representation in the simple blocks world is a powerful aid to acquisition. It would have been interesting to see if the program would be able to induce the action being performed by noticing changes in the environment, rather than having this information supplied as input.

2.1.3 Another Verbal Acquisition System

Salveter (1976) has described a model which will associate the surface representation of a verb with a corresponding conceptual dependency network. As input, the

program requires a Natural Language sentence and a set of environmental snapshots.

Salveter's concept of verb structure is similar to that of Schank's as embodied in Conceptual Dependency Theory. The first verbs that are acquired are the simpler, more primitive ones. Refinement of and extrapolation from these primitive verbs, eventually leads to the acquisition of the more complex verbs. In accord with this philosophy, is the use of Schank-like Conceptual Dependencies as a meaning representation for the verbs to be acquired.

The model was not designed to simulate human learning, but does try to be consistent with psychological data on the order that children learn verbs and the mistakes that are made in doing so. It is assumed that the initial state of cognitive development of the model is at the level of a 2-year old child. At this point, most of the primitive verbal concepts necessary to promote additional acquisition are available. Also assumed known are the concepts corresponding to physical objects and some grammatical knowledge along the lines of "actor-action-object". In addtion, the model is supplied with some built-in world knowledge.

Verbs in the model are represented by Conceptual Meaning Structures which, as was mentioned earlier, are similar to Schank's Conceptual Dependencies. Each structure consists of two parts; a set of case slots which describe

the noun concepts that can participate in an action, and the verbal effects, which describe the changes that take place in the environment when the action is carried out. These meaning structures are set up such that the meaning of one verb can also be a component of the meaning of additional verbs.

The environment the program operates in is that of a single room containing people, objects and referential locations. The environment at time Ti is described by a set of triples with each triple having the form,

(object relation verb)

Input to the program consists of a further set of triples ("a snapshot") at time Ti+1 and a Natural Language sentence describing the action that took place. Hence, events in the environment are represented in terms of state changes.

The first step in processing the input is to parse the input sentence to find the subject (actor), verb (action), objcts and unknown words. Also the features of the known words are located in the world knowledge base. To determine which action took place, the environment at time Ti is compared to the environment at time Ti+1. A list is made of all the triples in the first environment that are not in the second and vice versa. The program then tries to explain these differences by comparing the unmatched triples. If two of these unmatched triples match in every place except

for the value position then the likely event which took place is,

(object relation value)Ti-->(object relation new-value)Ti+1

The reason for this, is that it is more plausible for objects to change values than for values to change objects.

The information gained by "explaining" these differences is then used to locate a meaning structure that best accounts for the given event. If the surface verb is known then it's meaning structure can be directly retrieved. However, it may have more than one meaning structure associated with it corresponding to the different senses of the verb. These different senses of the verb are acquired when a retrieved meaning structure does not completely account for the changes in the environment. Thus, if more than one meaning structure is retrieved the model will choose the one that most closely accounts for the number of environmental changes. On the other hand, if the surface verb is unknown, then a meaning structure must be found that closely accounts for the environmental changes. This structure may currently be associated with another surface verb. Hence the difficulties in choosing a correct structure are that more than one structure can account for all the changes in the environment, or several may account for subsets of changes, or no unique meaning structure can be found at all.

The selection of a retrieved meaning structure activates one of five learning processes; confirmation, synonym, minor adjustment, major adjustment or definition creation. The type that is used is determined by the similarity between the changes in the environment and the constraints on the retrieved meaning structure.

Confirmation learning is applied if the program knows the verb and the retrieved meaning structure agrees with the input. For example, the verb "carry" could require as subject "male", object "toy" and action "identical location changes for both subject and object." If these restrictions are met, then corresponding frequency counts are incremented. These counts help determine "strangeness" when all restrictions are not satisfied. These counts are desirable in that they aid in the determination of whether conflicting information constitutes a strange case.

In synonym learning the program has no meaning structure associated with the input verb, but has retrieved one associated with another verb that accounts for the input sentence and the changes in the environment. The result of this process is that the meaning structure is now retrievable by another name.

Minor adjustment learning occurs when the closest meaning structure retrieved cannot totally account for the input. Hence the meaning structure is modified by changing its case restrictions. For example, if the mismatch occurs

on the "actor=male" restriction, (input is "female"), the world knowledge is searched to find the smallest superset of these instances, possibly resulting in the restriction being changed to "human".

Major adjustment learning requires structural changes to a meaning structure such as the addition or deletion of restrictions. The necessary information comes from the "explanation" of the differences found in the environment. As a partial example, consider the modifications of the structure for "carry" so that it can be applied to "throw". If "carry" has the restriction that the subject is in contact with an object at both beginning and end of an action, then this will have to be altered to reflect that contact is broken for "throw". Similary, since "carry" implies that the subject changes location with the object, this condition would have to be deleted since this is not the case for "throw".

In definition creation learning, the program has to create an entirely new meaning structure. The corresponding case slots are described by the classes to which the input words belong and a list of observed changes in the environment. In the types of learning that alter a meaning structure, the actual structure chosen depends on the number of changes that are necessary with each type only allowing a fixed number of changes.

There appear to be a number of difficulties or

weaknesses in the model. The first step in the processing of the input seems suspect in that the model parses the input sentence to obtain the subject, verb, objects and unknown words. This implies that the model has some built-in syntactic knowledge allowing it to detect which word is a verb, but not necessarily know its meaning. In addition, without knowing the meaning of a verb, the actor and object are somehow determined. Another point of contention arises with minor adjustment learning where a case restriction may be relaxed to account for a new word. This is a form of concept generalization which could lead to such lax restrictions on a meaning structure that many verbs may match it. This could result in the generation of a very general structure which may be bereft of information. To avoid such problems, it is probably necessary to design the world knowledge carefully. Since there does not appear to be any error recovery mechanisms in the model, it is probably essential to avoid the making of any errors at all costs.

The manner in which complex verbs are acquired from knowledge of primitive verbs is appealing. However, how effective this approach is in the acquisition of verbs, is as yet still uncertain.

2.1.4 Summary of Vocabulary Acquisition

The models which have just been examined have demonstrated some success in the acquisition of word meanings. Their greatest defect, which is difficult to measure, is that they operate in isolation from the other facets of language acquisition. Whether the same techniques or heuristics can perform as well in conjunction with the other learning processes is unknown.

What is significant, in terms of the current research, is the ability of the McMaster and King models to perform under conditions of unrestricted input. A similar association technique will be used in the model discussed in Chapter 3.

2.2 Structural Acquisition

In the following section three models are examined whose emphasis is mainly on structural acquisition. Siklossy's model is concerned chiefly with the discovery of mapping rules which will translate a semantic structure into a natural language sentence. In the vein of simulating human behavior, Anderson's model attemps to acquire a transition network grammar for natural language. Finally, Reeker's model explores the simulation of child language acquisition.

2.2.1 Second Language Acquisition

Siklossy's(1972) program ZBIE, was designed to explore at an elementary level, certain aspects of second language acquisition, yet does not attempt to model human learning behavior. The task put to the program is to express in natural language a situation described by a uniform, structured functional language which is Siklossy's version of a semantic referent. The philosophy of the program is based on Richard's(1961) "Language Through Pictures" series, with the functional language taking the place of the pictures. A given learning situation is represented by a functional language description and a natural language expression. Successive comparisons with other similar situations comprises the acquisition process.

The functional language (FL) is LISP-like in nature, with verbs and function words being treated as n-place functions, i.e. (F1 X1 X2 . . . Xn). For simplicity, inflections and articles are not considered and to improve the descriptive power the referents of pronouns are specified. Some example FL descriptions with corresponding natural language expressions are,

(be hat [of boy])
This is the boy's hat.

(q be book here)
Is the book here?

(be (in (and hat book) drawer))
The hat and book are in the drawer.

In Siklossy's program, second language acquisition is equated with translation. The internal structure used for translating FL structures into natural language is called a "pattern". If a FL structure matches a particular pattern, then that pattern's "translation rule" is activated to effect the translation. A pattern can be defined in B.N.F. as,

```
<pattern>    ::=  <p-list><d-list>
<p-list>     ::=  <set name><extractor>|
                       <set name><extractor><p-list>
<d-list>     ::=  <attribute><value>|
                       <attribute><value><d-list>
<set name>   ::=  A1|A2|A3 . . .
<extractor>  ::=  Y1|Y2|Y3 . . .
<attribute>  ::=  <internal symbol>
<value>      ::=  <list structure>
```

Hence a typical pattern might look like,

P0 = A1Y1A2Y2A3Y3; TR = ((Y1Y2)Y3)

This means that pattern P0 has as pattern list (p-list), (A1Y1A2Y2A3Y3), where the A's are set names and the Y's, the corresponding extractors. To match a pattern successfully each successive element in a FL structure must be an element of each successive set. If so, then the corresponding element in each extractor is taken to be part of the ensuing translation.

The translation TR is contained on P0's description

list (d-list) and has the form TR = ((Y1Y2) Y3). The translation rule describes the manner in which the natural language elements extracted from the pattern list (elements of the Y's) are to be rearranged to achieve the desired form.

The above process can perhaps best be seen by way of an example. If we take the pattern,

P2 = A4Y4A3Y3; TR = (Y4Y3)

where,

A4 contains "be" , and Y4 "This is"

A3 contains "boy", and Y3 "a boy"

and as FL structure (be boy), then since "be" and "boy" are both elements of A4 and A3 respectively, we then take the elements of Y4 and Y3 and order them according to the translation rule to obtain "This is a boy".

The routine that matches FL structures to patterns only uses set inclusion, and if necessary, this is performed recursively. Other forms of translation rules besides the one given above include,

P37 = A12Y12A2Y2A3Y3; TR=(Y2Y12Y3)
the order of the extracted elements is rearranged

P1 = A1Y1A2Y2A3Y3; TR = (Y2Y3)
a FL part is omitted during translation

P0 = A1Y1A2Y2A3Y3; TR = ((Y1Y2) Y3)
the grouped elements are translated together

In the last rule, the extracted elements corresponding to Y2

and Y3 are translated in the context of their appearing together, as opposed to singly which might lead to a different translation. This is followed by the translation of Y3. Other translation rules considered, but not implemented, are ones where the translation rule contains some constant string in natural language and ones where a functional element is used more than once.

To this point we have only considered ZBIE's data structures and some indication as to how the translation rules work. The program organization and operation is discussd next. ZBIE has two modes of operation, initialization and single sentence processing. In initialization, internal structures are set up and the first pattern is constructed. To construct this first pattern two situations expressed in both natural language and FL are required and these two situations must be sufficiently similar so that ZBIE can build the pattern. To be sufficiently similar there must be no complex FL structures and there must be only one element different in the same position in both the natural language and FL structures. Also, the differing element must occur at either the beginning or end of the sentence.

After initialization, single sentence processing can begin. As each new situation is presented, an attempt to match the FL structure to the list of patterns is made. Usually a complete match cannot be found and hence the

program looks for as good a fit as it can find. In doing this a set of pattern lists is stored in a pattern list holder. These pattern lists correspond to close matches to the given FL structure. A record is kept of how complete each match was, (in matching all components of the FL structure) and whether the match was total or partial, (a mistake encountered or not). Up to one mistake is allowed for any one substructure in a FL unit. Next the program does a translation of all the pattern lists starting with those that matched the closest.

If some FL unit cannot be translated a "Z" (unknown placeholder) is inserted and a consistency test on the translation is then made. There will be consistency if the Z's obtained in the translation can be replaced by non-empty strings in natural language in a unique, un-ambiguous way so that the translation matches the input. Pattern lists which do not pass the consistency test are discarded.

Next ZBIE starts processing the remaining pattern lists starting with those whose translation had the best fit to the input. This process involves replacing each Z with a natural language string. The corresponding FL unit is inserted in the corresponding set of the pattern under consideration, (the NL unit in the extractor set). If this can be done unambiguously the program can then add a new pattern to its internal structure. Insertion is ambiguous if some other element of the set (where insertion is taking

place) can translate the same FL unit into a different natural language string.

Since matching takes place with patterns in the reverse order in which they were created it is possible for a match and translation to take place before reaching an older pattern that would have performed a similar job. This results in some of the older patterns being no longer reached and is beneficial in that some of these older patterns may have incorporated errors.

As we have seen, ZBIE is able to improve its performance so that previous translating methods can be replaced by better ones. ZBIE tries to minimize the amount learned at each stage by using the maximum amount of information available from previous situations. The program will abandon the learning of situations if they become too "difficult", which is essentially a technique of "error-avoiding".

Siklossy, in summary, points out a number of faults in his system. The initialization stage is inflexible and must be performed correctly since all that is to follow depends heavily on it. The translation rules are not quite as general as they could be; no account is taken of prefixes and suffices, the means of expressing a concept is restricted by its translation rule. Also, as the number of patterns increase, so does the required search time for matching and unnecessary patterns are often created when the

context for the same verb changes. Two final important items are that a good teaching sequence is extremely important and that the system has limited error recovery.

2.2.2 Acquisition of Augmented Transition Networks

Anderson (1977) has developed a Language Acquistion System (LAS) which is designed to acquire Wood's-like (1970,1973) Augmented Transition Networks (ATNs). The intention was to develop a psychological model of human language processing, though the non-declarative or procedural aspects of language, such as question processing were not modelled. Actually, LAS like ZBIE, is best thought of as a model of second language learning in that all the concepts that are referenced in the sentences from which the model is to learn are known. Even though it was recognized that conceptual development is a prerequisite to grammar induction and that it continues with the acquisition of language, Anderson decided to restrict LAS to investigate only grammar induction. This separation of conceptual development and grammar acquisition is a common characteristic of most language acquisition models. Thus, LAS is expected to relate strings of words to corresponding semantic representations, but not to model how the semantics of individual words are acquired. Anderson claims that it is trivial to write a program that can acquire word meanings as well as grammar. To do so only involves saving those concepts that were in the semantic referent each time a

given word is used in a sentence and eventually set intersection will make it possible to identify the concept corresponding to the word.[1]

As input the program requires a natural language sentence, a semantic network representation and an indication of the main proposition of the sentence. The semantic repesentation used in LAS is based on Anderson and Bower's(1973) HAM memory system and a similar memory searching algorithm was also used. In acquiring the grammar the program induces word classes, rules of sentence formation and semantic mapping rules. The end result is a grammar which is capable of both sentence generation and sentence comprehension.

One of the main programs in LAS is BRACKET. It is responsible for taking an input sentence and semantic representation and producing a bracketing of the sentence which provides an indication of the sentence's surface structure. BRACKET assumes that there exists a constraint between possible surface structures and corresponding semantic structures which is called the Graph Deformation Condition. It is claimed that a sentence's surface structure can always be represented as a graph deformation (spatial rearrangement of links) of the semantic structure.

---

[1]This may be so in LAS' paradigm, but I do not believe it is as trivial as implied. At best, this would probably only apply to object concepts and not relational ones.

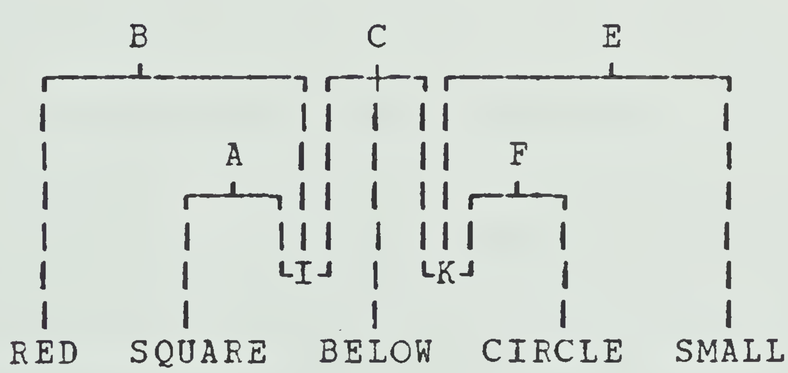
Certain word orders will be considered unacceptable ways to represent semantic intentions if in the rearrangement of links it is found that some links cross.

The first step in the bracketing process is to compute an intermediate structure called the Prototype structure. This prototype is simpler than the initial semantic representation but still contains the necessary information to calculate the surface structure of the sentence. The prototype is determined by comparing the semantic representation to the sentence to determine which elements of the representation are required. It will frequently be the case that the semantic representation will contain information irrelevant to the sentence in question.

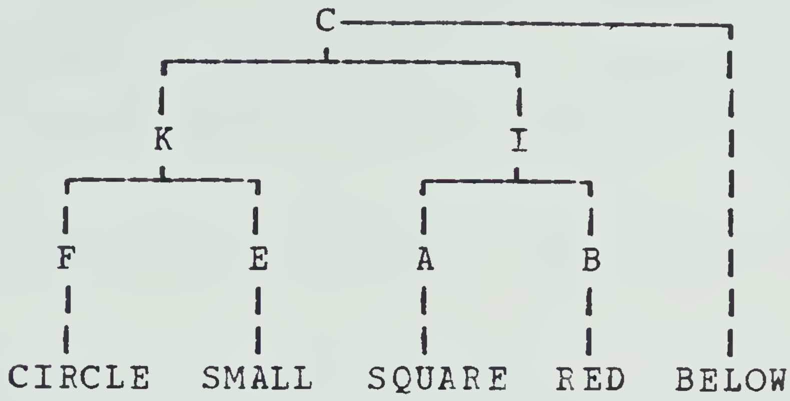
A possible prototype structure for the sentence,

THE SMALL CIRCLE IS BELOW THE RED SQUARE

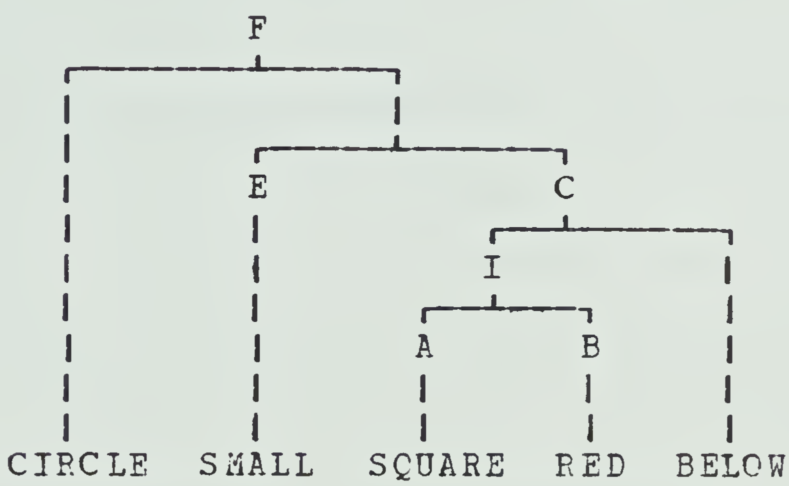
would be,



Once LAS has this prototype structure it then tries to find some graph deformation of it that will provide a tree structure connecting the content words of the sentence. Two possible deformations (surface structures) are,



and



The first is read,

THE SMALL CIRCLE IS BELOW THE RED SQUARE

while the other reads,

CIRCULAR IS THE SMALL THING BELOW THE RED SQUARE

The main difference in these two surface structures is that in the first, "C" is indicated to be the main proposition while in the second it is "F". It should be noted that in the prototype structure the arrangement of links as to "above", "right-of" or "below" is not significant, though in the resultant surface structure the spatial arrangement of links is specified. Anderson compares this graph

deformation condition to the sort of innate universals of language postulated by Chomsky(1965). The actual output of the BRACKET program for the first surface structure is,

((CIRCLE SMALL)(SQUARE RED) BELOW)

An important item of information in choosing the appropriate surface structure is the indication of the main proposition which Anderson likens to having a teacher direct attention to what is being asserted. While recognizing this as a rather strong aid to the program, it was used mainly for convenience. Anderson claims that a few heuristics would be sufficient to determine which deformation best describes the ordering of the words in the sentence and even if the occasional incorrect one was chosen, it would do no harm to the induced grammar. This would simply result in the addition of an alternate path through the grammar which does not affect the other parsing abilities.

As indicated above the actual output of BRACKET is an embedded list with the embedding reflecting the levels of the surface structure. Each level of bracketing corresponds to a single propostition which in turn is processed by a single ATN network. There does arise some difficulty with the insertion of non-function words into the bracketing due to the fact that there are no semantic features to indicate where these non-function words belong. The current heuristic is to place all these words to the left of a content word on the same level as the content word and to

close bracketing after this. Anderson claims that this works more often than not. A typical sentence where this will not work is,

THE BOY WHO JANE SPOKE TO WAS DEAF

It would be bracketed as,

((THE BOY (WHO JANE SPOKE)) TO WAS DEAF)

with "TO" not being part of the relative clause. The suggested solution is to tell the program what to do with the "TO". Anderson somewhat justifies this by pointing out that children when initially learning languge do not appear to pay attention to words like "to", "of", "about", "for", etc. They thus avoid the problem of deciding to which constituents they belong. The BRACKET program also has trouble with sentences that have discontinuous elements, as they systematically violate the graph deformation condition. An example sentence would be,

John and Bill, borrowed and returned,
respectively, the lawnmower.

Sentences such as these cannot be learned by LAS. However Anderson cites the fact that they are rare in English and are not dominant in other languages. He also says that they are not the sort of constructions that are easy to comprehend or acquire.

LAS makes a number of assumptions about noun phrase

structure. One, is that in all languages they serve the function of referring to objects and secondly they have the structure described by the following rewriting rules,

NP--> morphemes (MOD) noun morphemes (MOD)

MOD--> preposition (MOD)

Every noun phrase must have a noun which can be preceded by words like "a", "the", etc., possibly followed by an embedded list of prepositional modifiers; such constructs can also follow the noun. Since LAS knows which concepts can serve as nouns, identifying the noun becomes the key to unlocking the structure of the noun phrase. This knowledge is supposed to reflect cognitive development which LAS is not attempting to model. Anderson claims that this is not "cheating" since, "If one's goal is to produce a program that can learn natural languages and if natural languages all have this structure, then this criticism is clearly not valid." In other words whatever universals that are available will be used.

LAS is capable of expanding word classes within a network. As a simple example consider the following. A network for "JOHN KICKED MARY" would be,

START--∈N1-->S1--∈V1-->S2--∈N2-->STOP

where N1, V1 and N2 are the word classes that contain "JOHN", "KICKED" and "MARY" respectively. If the next input sentence is "FRED AMUSED JANE", LAS will discover that it

cannot parse this, since "FRED", "AMUSED" and "JANE" are not in the specifed word classes. This can be handled by expanding N1, V1 and N2 so that the second sentence will be accepted. Hence, from two input sentences, the network has been generalized to accept eight (2^3) sentences.

An important restriction on this expansion process is that the semantic actions associated with the network arcs are not altered. There is of course the danger of over-generalization in the formation of the word classes. Recovery would be easy in LAS if explicit negative information is given when mistakes occur, however Anderson cites evidence from Brown(1973) and Braine(1971) that this is not the case. This problem presently remains unresolved.

Another major process of LAS is the merging of networks. There is a continuing test made to determine whether one sub-network can parse the same phrase as another and if this condition is satisfied a further test is made to determine the amount of semantic overlap. If the overlap is significant then the networks can be merged. As with word class expansion, there is the danger of over-generalization.

LAS has shown the capability to learn a number of artificial and natural languages all of which have dealt with a two dimensional world of geometric objects. A point is made for restricting learning programs to such well-defined subsets of language since with open-ended programs it is often difficult to assess exactly which

aspects of language they are capable of handling.

To summarize, Anderson has developed a program which is able to induce word classes, a context free grammar and a set of mappings between surface structures and semantic propositions. The program depends strongly on a proper presentation sequence which does not have any grammatical errors and includes examples of all grammatical structures. Also, the semantics provided to the program must be constructed with care so as to avoid over-generalization as well as incorrect merging of sub-networks. These semantics must also satisfy the graph deformation condition. Anderson points out that LAS is a slightly defective model since, "it assumes more of the semantics of natural language than they provide." Also there are certain characteristics of natural language that cannot be handled by a context free grammar though these faults may be sufficiently minor that they can be dealt with by correcting procedures.

2.2.3 Problem Solving Theory

Reeker (1975,1976) has developed a model of grammatical acquisition which he calls Problem-Solving Theory. This model was designed for use in simulating child language acquisition. Input consists of a sample (adult) sentence and a semantic referent. Reeker is concerned primarily with structural or grammar learning and hence ignores word and concept acquisition.

2.2.3 Problem Solving Theory

Grammatical knowledge in the model is stored in a context-free phrase structure grammar. Even though context-sensitive grammars can generate languages that context-free grammars cannot, Reeker feels that this fact is less important for natural languages. The semantics of the individual sentences are represented by Reeker's Semantic Dependency Notation which has been influenced by generative semantics.

The major components of Problem Solving Theory are described below. The semantic representation and portions of the input sentence in consultation with the model's current grammar produce the model's version of the input sentence. The input sentence is "reduced" (simplified) and then compared with the model's sentence to determine a possible difference between the two. If there is a difference, a table of "connections and changes" is referenced to effect a change in the model's sentence which will bring it more closely in line with the input sentence. A grammatical change is also indicated and a new semantic representation for this grammatical change is determined.

The reduction process is based somewhat upon the fact that "a child will often vocalize an imitation of an adult sentence". These imitations tend to be in a shortened or reduced form, presumably because of short-term memory

limitations.[2] Currently a set of three empirically derived heuristics are used to obtain these reductions,

1. Eliminate pure function words and inflections
2. Eliminate any meaningless (to the program) words
3. Eliminate the initial portion of the sentence as defined by short-term memory

These heuristics apply as long as the observed sentences are "too long" for short-term memory. There are several aspects of the reduction process that remain to be determined, such as whether the heuristics are in fact correct and if so can more definite rules be found? Reeker notes that adults are quite capable of producing these child-like reductions and this may aid a child in acquiring language. Also, it was noted that different reductions can lead to different paths of acquisition and also to failure.

After obtaining a reduced sentence the model attempts to match it as closely as possible to a sentence generated by the model's current grammar. If this fails then additional matches are attempted with variations of the reduced sentence which could include deletions and permutations. It is required that the semantic representation of the grammar generated sentence be

---

[2]The psychological validity of this process is questionable. In a paper by Bloom et.al(1976) on Adult-Child Discourse it was found that at an early stage (Mean Length of Utterance 1-1.5) imitative speech accounted for at most 23% of the utterances for one subject and as low as 12% for another. When the MLU rose to 2.5-3 the highest recorded percentage was less than 10 for imitative speech.

identical to a subtree (preferably a whole tree) of the semantic representation of the reduced sentence. The actual steps involved in the matching process are,

1. parse the entire reduced sentence, or
2. match the "last" part of the sentence, or
3. match the "first" part, or
4. match with an internal word deleted, or
5. failure.

After the reduced sentence has been compared to the generated sentence, a table of connections and changes is consulted which will provide a change to be made in the model's grammar. It is assumed that only single lexical items can be added, replaced, deleted or permuted in the grammar. If there are elements of the reduced sentence which differ from the generated sentence by more than a single lexical item, then no learning takes place. The changes that can be made in the grammar can be characterized as follows:

Addition-the differing element of the reduced sentence is either prefixed/suffixed to, or infixed in, the corresponding grammar rule

Replacement-the differing element replaces an element of the corresponding grammar rule

Deletion-an element of the grammar rule is deleted

Permutation-two elements of the grammar rule are permuted

For Deletion and Replacement to apply, the semantics of both the reduced and generated sentences must be the same and must be preserved throughout the change. Additionally,

Replacement is only used if none of the other changes apply, and the semantics of the resulting change agree with the situational semantics. These conditions allow for recoverability of the original structure from the semantic representation.

After the surface structure grammar has been augmented it is necessary that the new structures be related to the proper semantics. When the new structure is formed it has associated with it a semantic repesentation which is essentially that of the current situation. For the Addition changes, if the newly created rule is not unique then a semantic consistency check is made. Success is necessary for the rule to remain in the grammar so as to preserve generalization possibilities. For the semantic preserving changes of Deletion and Permutation, this check is not necessary.

An example of the above processes follows. If we have as input sentence,

That's a big man.

with meaning

THAT-->BIG-->MAN

then a reduced form would be,

That big man.

If the grammar generated equivalent is

big man

then the difference would be

That--

requiring a change of

PREFIX That

The grammatical change required is to change "Class(big man)" to "Class(that)Class(big man)", which could be written as "$S \to M^2M^1N$." The new semantics would then be

$$
\begin{array}{c}
\text{Semantics}(S \to M^2M^1N) = \ldots \\
\text{Semantics(Class(That))} \\
\downarrow \\
\text{Semantics(Class(big))} \\
\downarrow \\
\text{Semantics(Class(man))}
\end{array}
$$

The generalization that takes place is fairly simple. If there are several occurrences of two consectuive elements in the grammar then they can be replaced by a single new element. A new rule is then added to the grammar relating this new element to the two old ones.

It is assumed that nothing is retained from a failed attempt at learning; if the attempt is successful, the structure is learned. Also, constant use may make access to a particular structure more automatic while other structures become used less frequently.

The results of sample test runs indicate that Reeker's program can allow for gradual expansion of an initial grammar.

2.2.4 Summary of Structural Acquisition

The detailed analysis of Siklossy's and Anderson's models has been discussed above. There are a couple of points that should be highlighted however, as they have some influence on the model to be discussed in Chapter 3.

A useful feature of both models is their ability to forget or lose incorrect or less useful fragments of the grammar they are trying to acquire. This forgetting occurs as a result of the erroneous information being no longer referenced. Perhaps a better method would be the outright removal of such information, but the idea that such errors need not prove fatal is important. The model in Chapter 3 incorporates a similar concept in the acquisition of its grammar.

The main weakness of both models arises when one considers their inflexibility of operation. Siklossy's model requires a rigid and carefully controlled early training sequence to have any hope of success. In Anderson's model, all input must first pass the graph deformation test to be even considered as data for acquisition. This leads to difficulty in handling any unexpected or novelty input as one might expect to encounter in the real world. To a certain extent, the model presented in Chapter 3 will attempt to overcome this problem.

Perhaps the most successful of the structural

acquisition models is Reeker's, though I have some reservations abouts its success as a simulation model. Also, it is not clear just how far Problem-Solving Theory can go in accounting for the continued growth of language. Because of the emphasis on child language acquisition, this model has not had much significant influence on my own research.

2.3 A Comprehensive System

In the following section a compehensive model of language acquisition is discussed. The term "comprehensive" should not be taken to imply sophistication, but rather, as an indication of the variety of learning tasks that are considered.

2.3.1 Child Language Acquisition

Itagaki (1976) has designed a model to supposedly simulate child language acquisition. The acquisition process consists mainly of the association of words with concepts and the derivation of semantic mapping rules. The initial conceptual knowledge assumed by the model is structured along the lines of a verbal case system, (Chafe, 1970). Each action concept has a number of case slots which coincide with the categories of non-action concepts. For example, the verbal structure for "GRASP" has slots for the categories "animate agent" and "object complement". The

model also has some knowledge as to which concepts can act as "agent", "object", etc.

Input to the model consists of three components which include, sentences derived from a toy world, corresponding referential meanings and an indication of attention points. The referential meanings are expressed as two or three episodic "pictures". As an example consider,

<u>sentence</u>

The big yellow house is now broken

<u>episodic pictures</u>

picture1: (BIG) (YELLOW) (HOUSE) (<u>WHOLE</u>)

picture2: (BIG) (YELLOW) (HOUSE) (<u>BROKEN</u>)

The underlined components are the attention points. It is from the referential meanings and attention points that a corresponding semantic structure is built. A deep structure for the above example is shown below,

```
BIG(size)---------| HOUSE--------------BREAK
YELLOW(colour)----|(object, patient)  (process)
```

The parentheses indicate the attributal categories of BIG and YELLOW, and the verbal category of the concept BREAK. For HOUSE, they indicate the relevant case relations corresponding to those required by BREAK. It is by comparing this derived deep structure to the given sentence that words and mapping rules are acquired.

The acquisition of words consists of manipulating a

concept's association list of candidate words. The manner in which these candidate words were selected was not mentioned. As soon as a 1-1 association is obtained between a concept and a word, the word is considered known. Candidate words can be eliminated from a concept's association list if they do not appear in a corresponding deep structure, e.g.

concept: "ball"
word list: (GRASP, SNOOPY, RED, BALL)
sentence: "Charlie possess a big ball"
deep structure: (POSSESS CHARLIE ((BIG BALL) (BLUE BALL)))

In this case GRASP, SNOOPY, and RED can be eliminated from "ball"'s association list. When a concept and a candidate word are considered associated, then that word can be eliminated from all other association lists. From the above example, the word "ball" can now be removed from all other concept association lists. A candidate word can also be heuristicaly associated with a concept. If the model has as input sentence, ". . . grasp ball . . .", with corresponding semantic structure, (GRASP . . . BALL), mapping rule, "agent complement" and "grasp" associated with "GRASP", the model will associate "ball" and "BALL" since they have the same conceptual role of "complement" in both surface and semantic structures.

The acquisition of mapping rules consists of finding

ordering regularities of conceptual categories such as "action complement". For the sentence, "Charlie grasp pyramid", with semantic structure, (GRASP CHARLIE ((BLUE PYRAMID)(SMALL PYRAMID))) and conceptual categories as below,

CHARLIE: animate agent

((BLUE PYRAMID)(SMALL PYRAMID)): object complement

grasp: action

the model can acquire the mapping rule "action complement" on the basis of the substring order "grasp pyramid". This mapping rule is then added to the model's conceptual knowledge.

Testing of the model consisted of presenting 17 sentences with corresponding referential meanings to the model two times. The model was able to correctly acquire 18 of 22 words and 10 of 16 possible mapping rules. The rather remarkable acquisition rate, considering the small amount of input, casts some doubt about the claim that the model simulates child language acquisition. Disregarding this point, there is still some contention as to the significance of the model. The model seems to be so highly biased towards correct learning such that it cannot help but succeed. Each learning situation consists only of the information necessary to understand it. That is, attention points are given rather than derived and knowledge of conceptual categories is complete. There was no provision

for the correction or handling of any errors, since conceivably the model does not make any errors. There was also no indication of the extensibility of the model to handle the acquisition of words without direct referents in the environment.

The use of two or three episodic referential meanings is a noteworty idea, particulary in dealing with the learning of process verbs, as well as gaining an understanding of tense. Also the derivation of a semantic structure from referential meanings is a useful and probably necessary heuristic for language learning.

2.3.2 Summary

Because of the numerous defects associated with Itagaki's model, it has not had much influence on my work. In the next chapter, a hopefully more significant comprehensive model will be presented.

Chapter 3

A NEW COMPREHENSIVE MODEL

The model described in this chapter is an attempt to ameliorate severval diverse tasks involved in computational language acquisition. One aspect of the model is the continuation of the basic philosophy of McMaster and King. That is, the model is designed to operate effectively without the aid of special training sequences or "expert" feedback. Because of this approach, the model has no error-avoiding or specified recovery techniques. Instead, incorrect or incomplete knowledge eventually is replaced by knowledge that is more complete or correct. Though the presence of "defective" knowledge probably slows the acquisition rate of the model, it does not completely inhibit its operation. By being less stringent in the monitoring of these errors the model benefits in terms of execution efficiency and in independence of operation.

As an extension to the work of McMaster and King the current model attempts to acquire word meanings that do not have direct referents in a given environment. These words fall into the linguistic categories of determiner, adjective, preposition, etc. Also, it is the aim of the model to make use of the word meanings that it has determined it has acquired to aid in the acquisition of

additional words.

Another distinguishing aspect of the model is the manner in which structural knowledge is acquired. The approach differs significantly from that of Reeker, Siklossy or Anderson and follows much the same procedure as is employed in vocabulary acquisition. Where the approach of the above models is slightly deductive in nature and precise in execution, the present proposal is perhaps more inductive and certainly less precise. The assumed "cognitive" abilities are also more general and maybe even more "natural". The process of structural acquisition is, in a sense, a form of pattern recognition which eventually results in the detection of common word orderings. In dealing with this task, the model makes only the basic assumptions that simple sentence stucture is built around objects, actions, and that the information conveyed by a sentence is parsed in a left to right manner.

Finally, the model also engages in a form of concept acquisition, or perhaps a better term would be concept generalization. In concept generalization, sets of concepts are combined in such a fashion so as to construct new sets which are more general than the original sets were. Though few other acquisition models incorporate such a feature, it was found that knowledge in this form is essential to the overall acquisition process. More often than not, the availability of such knowledge is usually assumed by the

other models.

To summarise, the model described in this chapter was designed to explore computationally three facets of language acquisition. These include the acquisition of words without direct referents, structural acquisition and concept generalization.

3.1 An Overview

Though the learning processes mentioned above can be classified as being separate, in reality they are heavily dependent on each other. For this particular model the dependencies are indicated below those illustrated earlier in figure 2.1. The net result of complete acquisition is a model that has the ability to handle both sentence generation and sentence comprehension.

3.1.1 Concept Generalization

Before any structural acquisition or concept generalization can take place, some word-concept associations are necessary. The easiest words to acquire are those with direct referents in the environment. As mentioned earlier, this model partially extends the work of McMaster and King and thus assumes that these words have already been acquired. From these initial word-concept associations, the model can then begin concept

generalization. This process involves the comparison of words at a conceptual rather than lexical level. Common concepts are retained, labeled and then used in further comparisons. As each new generalization is formed, a supposedly more general description of a word class is produced.

This procedure was significantly influenced by the work of Nelson (1974,1977) who showed how object definitions can be acquired through successive comparisons of situations in which a particular object occurs. Eventually, via a discrimination process, the possible actors, actions, properties and locations associated with a given object can be determined. In the current research, emphasis is restricted mainly to properties and locations.

3.1.2 Structural Acquisition

The word-concept associations and concept generalization knowledge which the model has acquired provide necessary information for the acquisition of structural knowledge. The process of recognizing possible sentence structures is essentially a pattern recognition problem; the model has to detect common word orderings. Since the model is designed to handle noisy linguistic input it is clearly impossible to allow for all unique sentence forms including non-standard syntax. Hence pattern matching at the lexical level is unmanageable and it is necessary to

match at a deeper conceptual level. Matching at the conceptual level should provide a higher success rate and this is why the word-concept associations are required. Also, the generalizations that have been formed allow for more successful matching since each successive generalization accepts a larger class of words. That is, the greater the degree of generalization, the less restrictions there will be on making a successful match.

3.1.3 Word-Concept Associations

Eventually the structural and generalization knowledge becomes one source of information for the acquisition of further word-concept associations; in particular those words without direct referents. The other source of information comes from the environment in which the model operates. For the most part, information for prepositions and determiners will come from the environment while for adjectives and adverbials the information will come from the generalizations and word-concept associations. For example, if the objects named in an input sentence have some spatial relationship (preposition) between them, then the model can identify those relational concepts which may apply to some unknown word in the sentence.

The general idea embodied in the acquisition of these unknown words is that they provide information regarding the uniqueness of the objects and actions already known by the

model. If the model can eliminate the common (in a generalized sense) concepts associated with a known word, then it will have some indication as to which concepts apply to the unknown words. As in McMaster, a weighting scheme is used to evaluate the probable associations.

While the above steps have been dealt with separately, the realistic view is that they occur together. Hence the model is often working with fuzzy, incomplete information which further complicates the learning process. It is no wonder that language acquisition for people is such a difficult and time consuming endeavour.

3.2 Basic Definitions

There are three categories of knowledge embodied in the model corresponding to word meanings, word orderings and environmental relationships. Word meanings and object relations are described by a simplified predicate calculus representation. Admittedly this simplified representation does not have the expressive power of semantic nets or conceptual dependencies, but it is probably sufficient for the problems explored by the model. Though the use of a more complete representation will eventually be required by the model, at this early stage the inherent complexities of such representations would only add unnecessary complications. The data structure for word orderings is a Woods-like (1970) augmented recursive transition network.

The justification for a transition network representation is that it provides a convenient structure for the overlaying of possible sentence forms.

3.2.1 Word-Meaning Dictionary

Each word known by the model has an entry kept in the model's dictionary. The form of each entry is,

("word" (F C1 C2 . . . Cn))

Each Ci has two parts; an attribute and corresponding value,

Ci = (Ai Vi)

A set of these attribute-value pairs comprise the essense of a particular concept. Each attribute can have a number of values which correspond to either a sort of "sensory primitive" or another attribute. The value may be left unspecified in which case a placeholder is used until it becomes necessary to fill the value position. There is also an indication of the function or word class to which the word belongs. This has the form,

F = (func Fi)

Initial values of Fi are OBJ (object) and ACT (action) corresponding to objects and actions already known. Additional values of F are internal names generated by the model as it acquires more words. This function concept enables the model to keep separate the different classes of

words which are necessary in the structural and generalization learning processes. A typical dictionary entry would be,

```
(pyramid1 ((func obj)(object physical)(location unspec)
           (state moveable)(shape rectangular)(size large)
           (color pink)))
```

Actually this representation is very similar to that of Quillian (1969) as can be seen by the two "different" representations of "block1" below,

```
(block1    ((func obj)(color red)(size large)))
```

and

```
                Block1
                  |
                  v
              < *   * >
                |   |
                v   |
             BLOCK  |
                    v
                ( *   * )
                  |   |
           ┌------┘   └------┐
           v                 v
      ( *       * )     ( *      * )
        |       |         |      |
        v       v         v      v
      COLOR    RED      SIZE   LARGE
```

As was indicated above, a concept is defined by a set of attribute-value pairs. Each of the attributes of a pair can themselves be a concept which in turn is defined by a set of attribute-value pairs. Because of this recursion in the defining of a concept, the term "concept" will be used interchangably for either a set of attribute-value pairs, or for a particular attribute-value pair.

3.2.2 Transition Network

The transition network consists of an ordered set of nodes linked by sets of ordered arcs. Each node corresponds to either a word, or a generalization of a class of words. The arcs describe node (word) orderings that the model has so far encountered. Each arc has a "use count" which indicates preferred transitions and hence word orderings. Also a "creation date" is kept for each arc and is used in a later "editing" phase.

A simple network fragment would be,

```
          P1      P2      P3      P4      P5
      ----------------------------------------------
R1   | •-W1>•-W2>•-W3-•    • • •
     |      ^    v
R2   | •-W4-•    •-W5>•-W6-•    • • •
     |                v
R3   |                •-W7>•-W8-•    • • •
```

where the Wi correspond to words or generalizations and the Pi and Ri are simply positional markers.

An internal description of the above, without the utility information required by the implementation, would be,

```
network = (P1 P2 P3 P4 P5)

P1 = (P1R1 P1R2)

     P1R1 = (W1 [data on W1] (W2))
     P1R2 = (W4 [data on W4] (W2))

P2 = (P2R1)

     P2R1 = (W2 [data on W2] (W3 W5))

                    . . .
```

Each PiRj contains the given node, data on what is known about that node and a list of other nodes to which the current node is linked.

Eventually the model will be able to detect those node and arc combinations corresponding to common word orderings. Periodically the older, and less frequently used components of the network are removed, resulting in a somewhat simplified network.

3.2.3 Environmental Relationships

The physical relationships among objects in the environment at a given point in time are described by a relation dictionary. This relation dictionary has a similar structure to that of the word-meaning dictionary, but its contents vary from scene to scene whereas the word-meaning dictionary remains relatively fixed; the latter is altered only when newly learned words are added to it.

The general form of an entry in the relation dictionary is given by,

(object:i (R1) (R2) . . . (R-n))

where each R:i = (relation:i object:j). An example would be,

(block1 (beside pyramid2) (supports block2))

This dictionary was not designed to reflect any insights into the structure of such relational knowledge, but instead to provide a convenient means of referring to the relevant relationships when necessary.

3.3 Concept Generalization

Concept generalization occurs as a result of the comparison of the associated concepts of similar words. The words that initially take part in concept generalization are the objects and actions assumed to be known by the model. Eventually, as additional words are acquired, they too take part in the generalization process.

The model has been designed to explore the effects of this generalization process in two contexts. The first involves the formation of generalizations prior to attaching any significance to word orderings. This procedure is discussed below in section 3.3.1. An alternative approach is to immediately begin to generate word generalizations and attempt to interpret word orderings together. A discussion of this process is included in the section on building the

network, (3.4.1). In Chapter 4 an evaluation is made of these two approaches.

3.3.1 How Comparisons are Made

The initial comparisons which are performed take place in a "skeletal" network. This skeletal network consists of nodes corresponding to the known words and their generalizations, as well as "filler" (empty) nodes for the unknown words. The use of filler nodes allows the model to ignore information irrelevant to the formation of generalizations. An initial input sentence such as "The large green pyramid behind . . ." would be represented as,

•-filler-•-filler-•-filler-•-pyramid-•-filler . . .

The presence of arcs in the skeletal network during this generalization process is not significant since at this time the model is not concerned with word order.

Additional input sentences are processed by the model one at a time, word by word, in a strictly left to right manner. Selection of available nodes in the network for comparison also proceeds in a left to right manner with no looping back allowed. As each word in the input sentence is examined, a search starting from the current position in the skeletal network is initiated in an attempt to locate a "suitable" node for comparison purposes. If the current input word is unknown to the model then no meaningful

comparisons can be made. The first filler node located in the search is assumed to match the unknown word and processing continues with the next word in the input. However, if the word is known to the model, then the model attempts to locate a non-filler node which has concepts similar to those of the input word. At this point the only differentiation among sets of concepts is whether they belong to objects or to actions. That is, all words corresponding to objects can only be compared with nodes corresponding to objects and similarly words corresponding to actions can only be compared to nodes corresponding to actions. If no such suitable nodes can be found, for either known or unknown words, or if there are no nodes left in the network, then nodes corresponding to the remaining words in the input sentence are simply appended to the end of the network. The details of the comparison process and the possible resulting generalizations are discussed in section 3.3.2 below.

A successful comparison is one that may lead to the formation of a new generalization. When such a comparison has been made, a check of the previous generalizations made by the model is performed to determine if there already exists a generalization which can account for the results of the comparison. If such a generalization is found, it replaces the corresponding node in the network; if not, a new generalization is constructed by taking the results of the comparison, labeling it and entering this new

generalization in the corresponding place in the network. The reason for the check of the old generalizations is that the comparison of different words can lead to the formation of identical generalizations, and there is no need, nor is it desirable, to keep the same generalization around under two names.

It should probably be noted that in the attempted comparisons described above, not all possible comparisons were made. This is a result of stepping through both the input sentence and skeletal network in a left to right manner. In a sense, this embodies a "laziness" approach. This means that the model is trading off some information (missing a possible useful generalization) for processing efficiency (far fewer comparisons are attempted). This might appear to be detrimental to the eventual knowledge that the model is to acquire, but it need not be the case. The model has been designed such that it builds on the knowledge it currently has, but alternatively it can continue to operate with incomplete knowledge, or even if knowledge has been lost (see section 3.4.2). Possible information that has been missed or lost can always be assimilated at a later date. The experiments in Chapter 4 explore to some extent the viability of such a philosophy.

3.3.2 <u>Types of Comparisons</u>

The comparisons that are made between an input word and

a node in the network take place on two levels; lexical and conceptual. The lexical comparison always takes precedence over the conceptual comparison since, if it is successful, no generalization can take place. A successful lexical comparison simply indicates that both node and input word are identical and hence, there is no new information to be gained. In this case processing continues with the next word in the input. On the other hand, if the lexical comparison fails, then an attempt is made at a conceptual comparison. This involves comparing all of the associated concept attributes of the input word to the corresponding ones of the node. Before going on to the details of the comparison, perhaps it should be reiterated that the actual comparisons are performed on concept attributes and not concept values, or combinations thereof. Concept values are really only significant when the model focuses on a particular exemplar, and not on a generalized class. It is for this reason that when a new generalization is formed, those concepts differing in the value position have an "unspecified" token inserted. If the values happen to be the same, then they are retained. The "unspecified" token is really only a type of placeholder which can be filled in when a concept is instantiated.

There are three major results which can arise from the comparison process: 1) all concept attributes match, but the associated words are lexically different; 2) a subset of concept attributes were matched and 3) all concept

attributes of a generalized node were matched. The significance and examples of each of these three cases is discussed below.

In the case where all of a node's concept attributes were matched by those of an input word, the model will attempt to form a generalization only if the node is not already generalized. This situation corresponds to the discovery of a set of common concept attributes which have different lexical representations. The formation of a generalization reflects the acquisition of this item of knowledge. As an example, suppose we have for a node,

```
(block1   ((func object)
         * (object physical)
         * (location unspecified)
         * (state moveable)
         * (shape rectangular)
         * (size large)
         * (color pink)))
```

and for an input word,

```
(pyramid2 ((func object)
         * (object physical)
         * (location unspecified)
         * (state moveable)
         * (shape pointed)
         * (size small)
         * (color yellow)))
```

then the matched concept attributes are indicated by "*'s" and the resulting generalization would be,

```
(<thing1> ((func object)
           (object physical)
           (location unspecified)
           (state moveable)
           (shape unspecified)
           (size unspecified)
           (color unspecified)))
```

The second possible result of the comparison process is that not all of the node's concept attributes were matched. This situation corresponds to the model's discovery of a common subset of shared concept attributes. The formation of a generalization in this case reflects somewhat the identification of core concepts, or concepts salient to a word's definition. To see this, suppose we have for a node "<thing1>" from above,

```
(<thing1> ((func object)
         * (object physical)
         * (location unspecified)
         * (state moveable)
         * (shape unspecified)
           (size unspecified)
           (color unspecified)))
```

and for input word,

```
(table    ((func object)
         * (object physical)
         * (location unspecified)
         * (state immobile)
         * (shape rectangular)))
```

The resulting generalization formed in this case would be,

```
(<thing2> ((func object)
           (object physical)
           (location unspecified)
           (state unspecified)
           (shape unspecified)))
```

The final possibility resulting from the comparison process occurs when all of a node's concept attributes were matched and the node already is generalized. This case simply reflects an instantiation of the node so that no generalization can possibly be formed. For example, if we have for a node "<thing2>" from above,

```
(<thing2> ((func object)
         * (object physical)
         * (location unspecified)
         * (state unspecified)
         * (shape unspecified)))
```

and for input word,

```
(pyramid1 ((func object)
         * (object physical)
         * (location unspecified)
         * (state moveable)
         * (shape pointed)
           (size large)
           (color pink)))
```

then since all of "<thing2>'s" concept attributes were matched and it is already a generalization, the model can do nothing but note this fact and continue with the next word in the input.

3.4 Structural Acquisition

The model incorporates three phases of structural

acquisition which include, 1) additions to the network, 2) editing, and 3) restructuring. Adding to the network is an on-going process while editing and restructuring occur only periodically.

3.4.1 Building the Network

Construction of the transition network begins after a period of concept generalization as described above. It is not assumed that concept generalization is now complete and in fact, this process continues as the network is built. As before, processing of an input sentence and movement through the network is performed from left to right.

For a given input word a search is initiated from the current position in the network in an attempt to locate either a lexical or conceptual match. If such a match is made, the matched node has a usage count incremented. In addition, if the match was conceptual, then there is a possibility of a new generalization being formed which results in the old node in the network being replaced with the new generalization. Finally, if there happens to be a link from the previous position in the network to this matched position, then the usage count of this link is incremented. If there was no such link then one is added with a usage count initialized to 1. A special case arises if there was no previous position, that is, the current input word was the first of a sentence. To handle this the

model adds a special "->" node in the first position of the network and then links it to where the input word was matched. Before going on to process the next input word, network position pointers are reset.

If no match was found in the network for the input word, then a new node must be added at the point where the search was initiated. In the case where a filler node remains from the skeletal network, the input word can then replace it, otherwise a new node is simply added where space permits. This new or replaced node has a usage count initialized to 1 and also a creation date associated with it. The same procedure as above for adding a new link is then performed and finally, the network position pointers are reset.

To see this construction procedure in action, consider figure 3.1. The first line is the skeletal network generated by the following three sentences,

Beside the yellow pyramid1 is a block2
The block1 is red
The block1 and pyramid2 are small

"<THING1>" represents a generalization formed by "pyramid1" and "block1" while "<THING2>" is a generalization of "block2" and "pyramid2". The following three lines in the figure show the network changes as the three input sentences are once again processed.

For the first sentence the words "Beside", "the" and

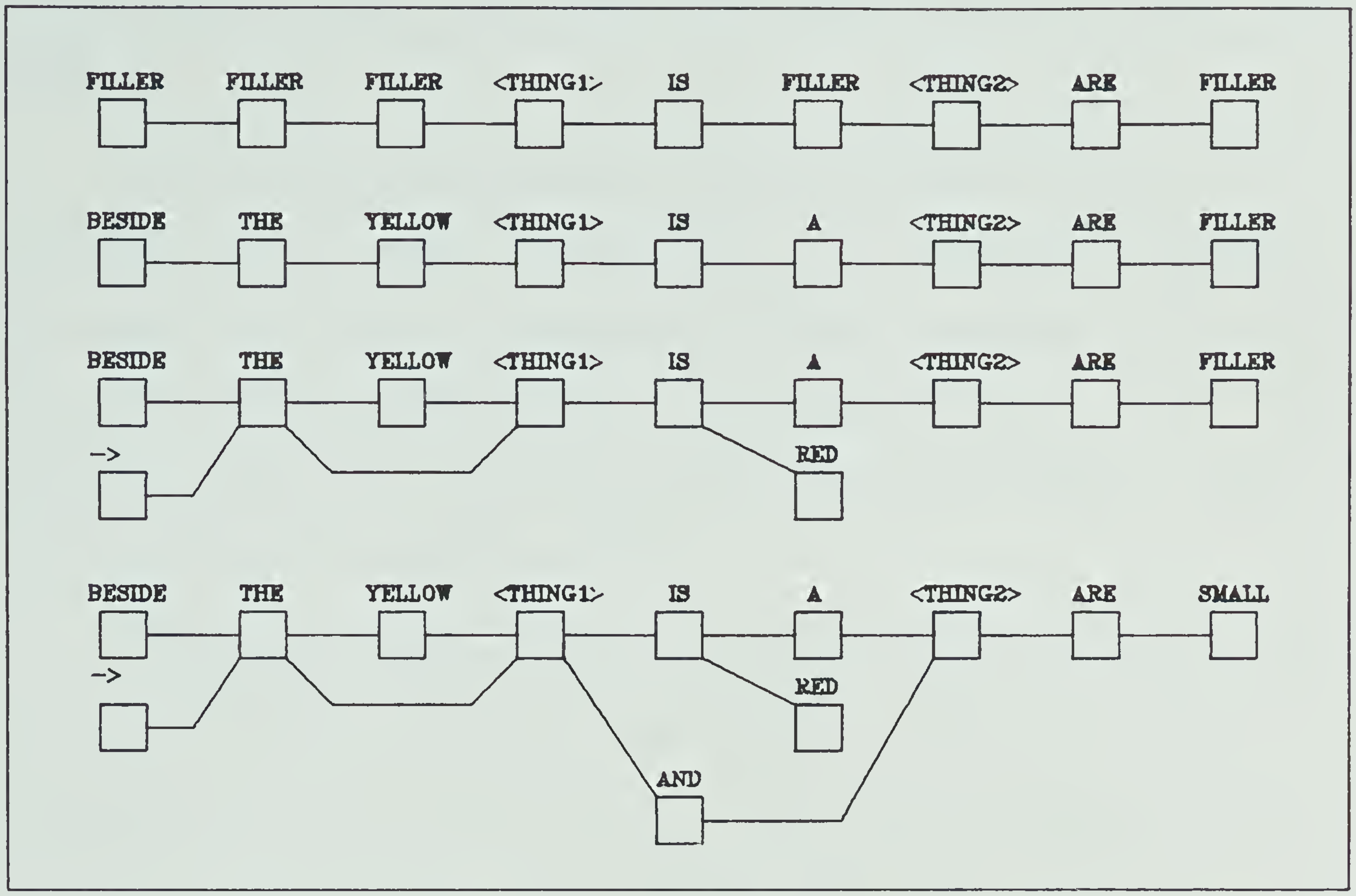


Figure 3.1

"yellow" have nothing to match in the network so they simply replace filler nodes. "Block1" and "is" are then matched by <THING1>" and the "is" node respectively. Next, "a" replaces a filler node and "block2" matches "<THING2>".

"The" in the second sentence matches the "the" node in the network but when checking for a link it is found that a "->" node must be added in the first position of the network. "Block1" matches "<THING1>" and since there is no link from "the" to "<THING1>", such a link is added to the network. Next "is" matches the "is" node as before, but there is no match available for "red". Since there is no filler node in the current position to be replaced, the

model simply adds a new "red" node and links it to the "is" node.

In the final sentence, "the" and "block1" match existing parts of the network but a new node and link must be added for "and". "Pyramid2" matches "<THING2>", but this necessitates the addition of a new link from the "and" node. Finally, "are" will match the "are" node and "small" will replace the final filler node.

As was mentioned above, the model keeps an account of the usage of the nodes and links in the network. This information is used to identify the preferred transitions through the network. The creation date that is associated with a node aids in the identification of network components that appear to have a low usage which is due only to their recent creation. These usage counts and creation dates are considered when the model enters a network editing phase described below.

One motivation for conceptual generalization was to provide a higher success rate in the matching of known words to the network. By having a generalized node accept several known words the need to add new nodes to the network is reduced. Ideally this will provide for a slower growth rate in network size and accordingly faster processing time. The effect of such a technique is explored further in Chapter 4.

3.4.2 Editing the Network

One motivation for the formation of conceptual generalizations was to slow the proliferation of nodes and arcs in the network. This motivation is based entirely on grounds of efficiency, whether in the matching of input sentences to the network or in the identification of common word orderings (to be discussed later). Despite using such a technique, it is still evident that there will arise certain parts of the network that are infrequently used and hence unnecessary for the early acquisition of common words and word order. Yet, since they are part of the network, they will still be considered (usually unsuccessfully) when a sentence is matched to the network. For this reason it was decided to periodically edit from the network these uncommon components. The problem of when such editng should take place remains unresolved. It is not known whether the editing should occur concurrently with the building of the network or whether it should occur on a "suitable" periodic basis. For experimental purposes, it was decided to edit the network after each set of input (see Chapter 4). Editing of the network involves the removal of those nodes and arcs which have a usage count below a given level. The creation date of a component is also taken into consideration in this decision process since a node's low usage count may be due entirely to its having just been created. Once again, optimal values for these "cut-off" points are unknown.

It should be clear that whatever efficiency is gained by dealing with an edited network is somewhat offset by the information that is lost. Whether this trade-off of information for efficiency is profitable is yet another unresolved question. However, the loss of such information need not be irrevocable as it can be reacquired at a later date. It may also appear that if information is lost at one point due to low usage, then it will probably be lost again in a later editing phase. The inherent implication is that the criteria for the removal of nodes and arcs will probably have to change with the growth of the network. In the early volatile growth of the network, the criteria should probably be rather severe, yet when the network stabalizes to a certain extent, such criteria should be relaxed. This relaxation of the removal criteria reflects to some extent the focus of the model shifting from a global point of view of sentence structure to a more narrow and local view. Those components that are repeatedly removed in the early stages will now have a better opportunity of remaining in the network. Despite the risk of being repetitive, it must be stated that this procedure for changing removal criteria is yet another unresolved question.

As an example of this editing process, consider the first line in figure 3.2 which reflects the structure of the sentences,

The red pyramid is large
The pyramid, which is red, is large

Clearly, the first sentence reflects the more common early

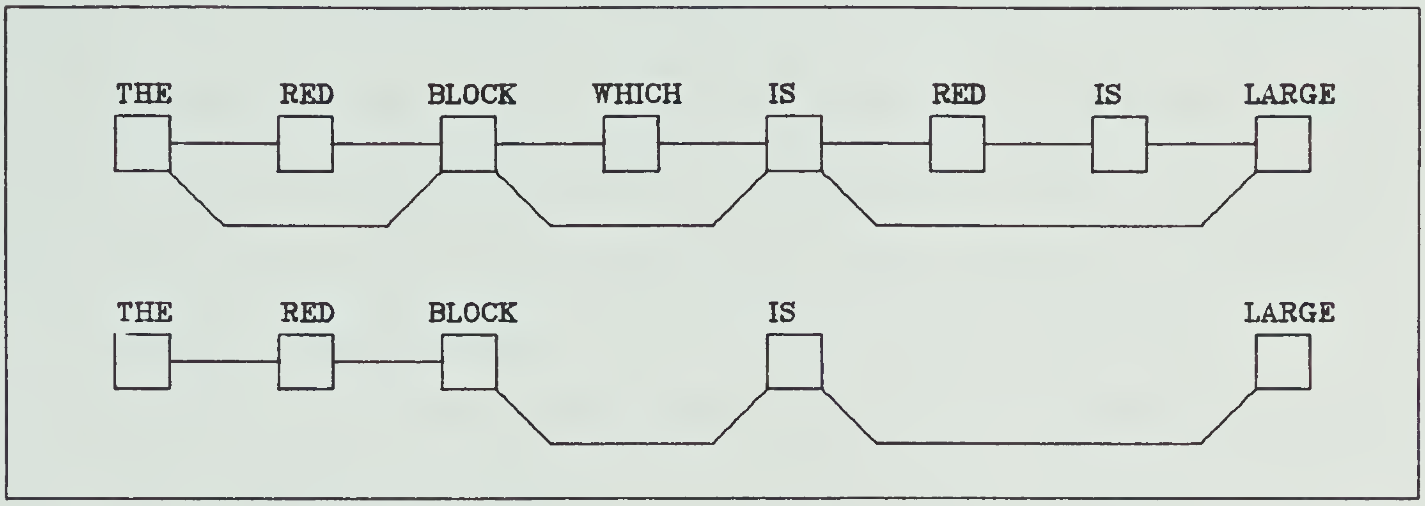


Figure 3.2

word ordering and the edited network would then be similar to the second line in the figure.

It should be evident from the above that there are a number of "fuzzy" areas in the editing procedure. A further discussion of how the "unresolved" issues can be resolved and their likely resolutions is taken up in Chapter 5.

3.4.3 Restructuring the Network

The restructuring of the network is essentially a type of syntactic generalization. Periodically, the model will attempt to identify the occurrence of identical node combinations throughout the network. If such combinations of nodes are found, then they are replaced by one common group of nodes. This syntactic generalization can occur

repeatedly thus resulting in the network taking on a recursive nature.

The impetus for this syntactic generalization is based mainly on considerations of space efficiency. In addition, it was decided that when these generalizations were found, they would be treated slightly differently from the rest of the network in that they would not take part in the usual structural acquisition. In a sense, these isolated components have been restricted from further growth and for an input sentence fragment to match them, it must do so successfully throughout the generalization. The reason for the isolation of network components is that once the model's criteria for the acquisition of knowledge of word order has been met, it should not be necessary to continually modify such knowledge. Inherent in this process, once again, is a trade-off between processing efficiency and the possibility of information loss. Information can be lost if the components of the syntactic generalization have not yet been fully generalized. If this is the case, the syntactic generalization will not match as many sentence fragments as it possibly could and the model will still have to generate suitable conceptual generalizations elsewhere in the network. The gain in processing efficiency arises from the localization of a (hopefully) significant accumulation of knowedge.

Even if the model should make an imperfect syntactic

generalization (i.e., one that is infrequently matched) this need not be disastrous. The usual editing processes described earlier will ensure that such a generalization will be removed due to low usage. Since this is a rather wasteful procedure, the model will not attempt to peform any generalization until the network has stabilized to a certain degree. An adequate means of identifying stability in the network remains to be determined. As with the editing procedure, the identification of the point when this syntactic generalization should commence is not fully known. I would expect however, that there is probably some connection between the changing of criteria for editing and the beginning of the formation of syntactic generalizations.

The actual details of selecting node combinations as suitable candidates for syntactic generalization are discussed next. Instead of allowing any node combination to take part in a generalization, only those combinations that have as one of their corresponding components an object, action or previous generalization are considered. Here the model is attempting to localize the generalization around the important concepts of objects and actions such that it may be possible to discover the equivalents of noun and verb phrases. There are also some search efficiency considerations. If there are n node combinations in the network, then for one such combination there are n-1 comparisons necessary to determine if another identical combination exists. For each of the remaining node

combinations a similar condition can arise resulting in $(n^2-n)/2$ comparisons. In the first experiment in Chapter 4 it was found that after processing only 30 input sentences, the resulting network contained 62 node combinations. If unconstrained node combinations were to be allowed in syntactic generalization then this would necessitate 1891 comparisons to be made.

After the model has constructed a list of candidate node combinations, it checks the highest usage of each combination to see if it is "sufficient" to be included in a generalization. If the node combination passes this test, then all examples of it in the network are replaced with a link to a single copy of the node combination.

As an example, consider the network fragment shown in the first two lines of figure 3.3. After making the necessary checks throughout the network the model will have determined that the combination "<GEN1>---<GEN2>" is suitable for syntactic generalization. It then replaces all occurrences of this combination with a branch to a single copy; in our example, at "SYNGEN1". This is shown in the last three lines of the figure.

3.5 Word-Concept Associations

So far the discussion has dealt mainly with the acquisition of knowledge related to word orderings. The following section describes how the model attempts to

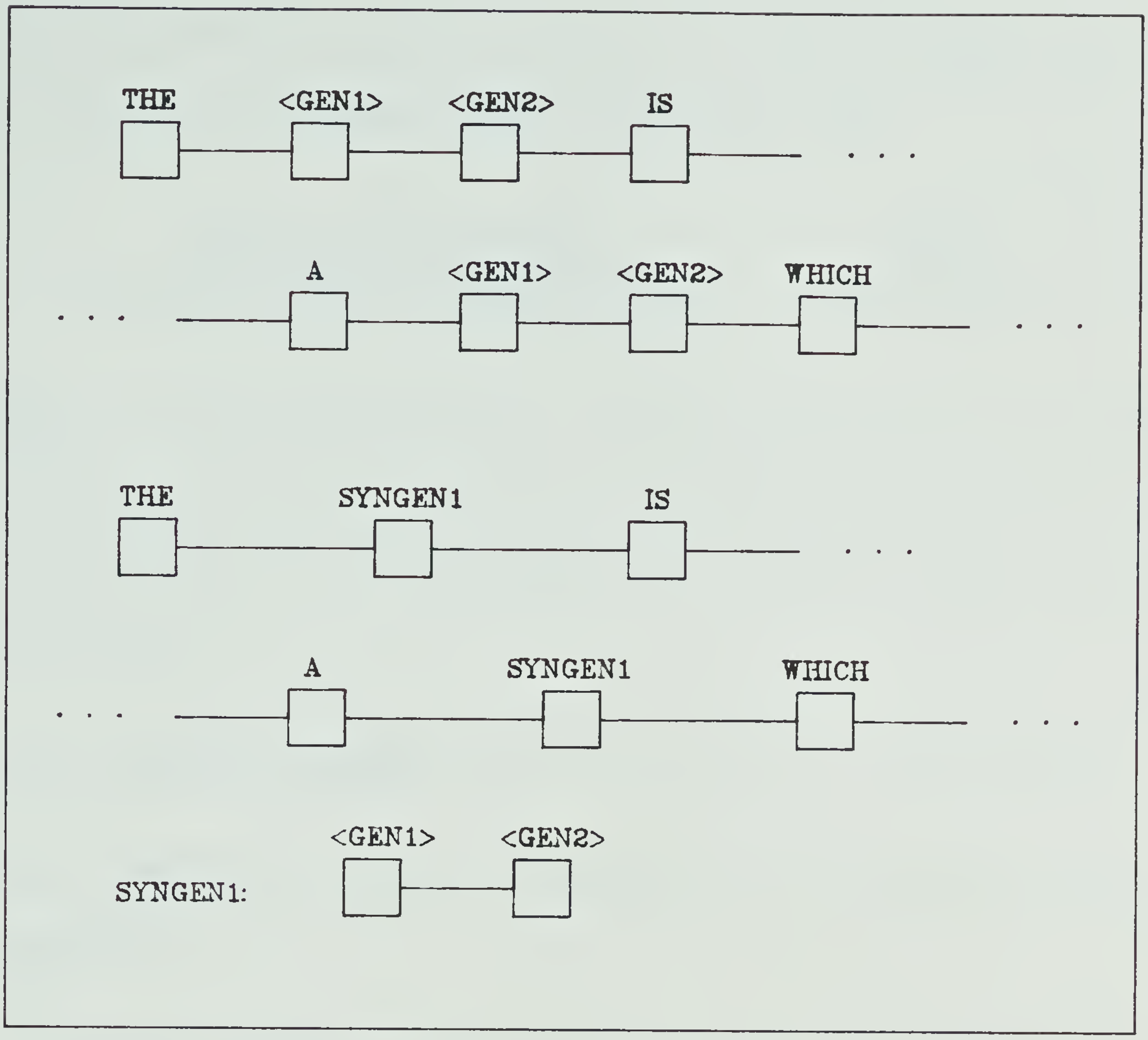


Figure 3.3

acquire word meanings. It should be remembered throughout this section that the focus of word acquisition is centered on those words which do not have direct referents in an environment. It is assumed that these words are either further descriptions of the known objects and actions, or that they provide the means by which words are related.

3.5.1 Input-Sentence Segmentation

The first step in processing an input sentence for meaning acquisition is to segment it into linear groupings of known and unknown words. Each group contains at least

one known word and possibly several unknown words. The three possible segmentations considered are,

unknown(s)-known(s)	(U-K)
known(s)-unknown(s)	(K-U)
known(s)-unknown(s)-known(s)	(K1-U-K2)

Inherent in this segmentation is the implication that there is a basic, or kernel structure of language that facilitates its acquisition. For instance, it is more likely that in the sentence,

The pyramid <u>beside</u> the block is . . . (K1-U-K2)

the word "beside" will be recognized as a relation than in the sentence,

<u>Beside</u> the pyramid is the block . . . (U-K)

where it might be taken as an object modifier.

Therefore, in either the U-K or K-U case, the model assumes that the U's are modifiers of the given K's. For K1-U-K2 groups the U's may be modifiers, relations or simple connectives.

3.5.2 <u>Selection of Candidate Concepts</u>

The segmentation procedure above labels U-K, K-U groupings as Case1 and K1-U-K2 groupings as Case2. The details relating to the processing of these cases is given in figure 3.5 with the corresponding terminology being

```
                         Terminology

    U     -- a list of one or more unknown or learned words,
               none of which is an object or action.
    U?    -- an unknown word
    U1    -- a word previously unknown, but now considered
               known by the model
    CC    -- candidate concepts for U?
    K     -- a list of one or more known words, (K1, K2)
   C(K)   -- concepts associated with the known words
   G(K)   -- generalized concepts of a known word
   C(E)   -- associated concepts in the environment
   C(U1)  -- concepts associated with learned words
  L(•••)  -- length of a list
    R     -- some relation
    +     -- add to a set
    -     -- delete from a set
    ∅     -- the null set

                       U? & [U1] ∈ U
```

Figure 3.4

defined in figure 3.4. A verbal description of the Concept Selection Algorithm follows.

Initially in Case1 there are no candidate concepts (CC) for the unknown words (U) in the current group being processed. The candidate concepts are then set to those concepts associated with the known words (C(K)) in the group as defined by the model's dictionary. This list is then reduced by removing the concepts associated with the generalized sense of the known words (C(G(K))) as defined by the generalizations in the network. The effect of these first two steps is to isolate those concepts that refer to the uniqueness of the known words. Next, the candidate concept list is augmented by whatever environmental concepts

Concept Selection Algorithm

Initially

CC <-- ∅

Case1

CC <-- CC + C(K)
CC <-- CC - G(K)
CC <-- CC + C(E)

if { ∃U1 | U1 ∈ U } then ∀ U1 do CC <-- CC - C(U1)

if L(CC) ≥ 1 then ∀ U? do U? <-- U? + CC

OR

Case2

if { ∃R | K1 <= R => K2 } then

CC <-- R
if L(U) = 1 then U? <-- U? + CC
else Case1 using K2

else

CC <-- (cnj unspecified)
if L(U) = 1 then U? <-- U? + CC
else Case1 using K2

Figure 3.5

that are present. Typically these concepts (which are explicitly given to the model) correspond to the linguistic categories of referent (direct or indirect) and number, (singular or plural). At this point, no further concepts are added to the candidate list. If however there exist some words considered learned by the model (U1) in the

group, then the candidate concepts undergo one further refinement. This involves the removal from the candidate concept list of all those concepts associated with the learned words. This is done since it is not likely that the same concept would be associated with more than one unknown word in a particular grouping. Finally the model is able to update its knowledge of the unknown words by incorporating the candidate concepts into the unknown word's association list. If a concept is already associated with an unknown word then a count is simply incremented, otherwise the new concept is appended to the unknown word's list. Processing then continues with the next group.

As an example, if for a Case 1 grouping we have,

((The large pink block1) Case 1)

then the following steps will show how candidate concepts are selected for the unknown words in the grouping. First,

CC <-- ∅

then the concepts associated with the known words in the group are added,

```
CC <-- CC + ((object physical) (location unspec)
             (state moveable) (shape rectangular)
             (size large) (color pink))
```

Those concepts associated with the generalized sense of the known words are next deleted,

CC <-- CC - ((object physical) (location unspec)
(state moveable) (shape rectangular))

and then those concepts derived from an examination of the environment are added,

CC <-- CC + ((reference direct) (number singular))

If there are any learned words in the grouping then the concepts associated with them are deleted. For illustrative purposes, suppose that "the" has been learned by the model,

CC <-- CC - ((reference direct) (number singular))

The net result of the above is that,

CC = ((size large) (color pink))

The association lists of "large" and "pink" are then updated to reflect the fact that the concepts of CC are possible candidates for their meaning.

Case2 differs essentially from Case1 in that it allows for the possibility of relational words and grammatical connectors. As before, the candidate concept list is initially empty. The model then investigates the environment to determine if there exits some relation between the known words in K1 and in K2.

If such relations are present, they are then added to the candidate list. In the situation were there is only one unknown word, its association list can be immediately

updated with the relations supposedly corresponding to this unknown word. Otherwise the model follows the same steps as in Case1. Where no relation was found by the model in the environment, it is then assumed that a conjuction or interjection may be present. As before, if there is only one unknown word in the grouping, its association list is immediately updated. If this is not so, then Case1 is used. Since Case1 assumes the presence of only one known word group, one of the known groups in Case2 must be dropped when a transfer to Case1 occurs. This group is K1, the first group, since the model assumes that language processing is essentially a left to right procedure. That is, it is more likely that the unkown words further describe the following known words, rather than the preceding ones.

As can be seen from the above, the Selection Algorithm is relatively simple and straight forward. It was designed in such a fashion so as to assume as little inherent linguistic knowledge as possible. The assumptions made are that language is usually processed left to right, and where there are two or more known word groups, relations or conjuctions may be present.

3.6 Evaluation of Associations

The associations made by the Selection Algorithm are periodically evaluated by McMaster's function which was described in the earlier section on his research. The

effect of this function is explored somewhat in a section in the following chapter.

Chapter 4

EXPERIMENTAL RESULTS

To test the validity of the model, five related experiments were run. The first experiment was to verify that the model could in fact increase its vocabulary as well as construct a rudimentary grammar. The next three experiments were designed to determine the relative effectiveness of various components of the model. The final experiment was designed to expand on the scope of the previous experiments. The model was implemented by use of an interpreter MACLISP system running under the Michigan Time-Sharing system (MTS) on an Amdhal 470/V6.

4.1 Experiment 1

While experiment 1 is essentially a test of the model's ability to produce some meaningful results, it is also a test designed to make the most use possible of concept generalization. The contention is that concept generalization is a necessary prerequisite for both vocabulary and grammar acquisition. Hence, before the model makes any attempt to expand either it's vocabulary or grammar, it is involved in a period of concept generalization with the words it is assumed to know. Following this the other learning processes are activated

through several iterations of a given input.

The particular blocks world used in experiment 1 is shown in figure 4.1. All the objects shown are assumed to

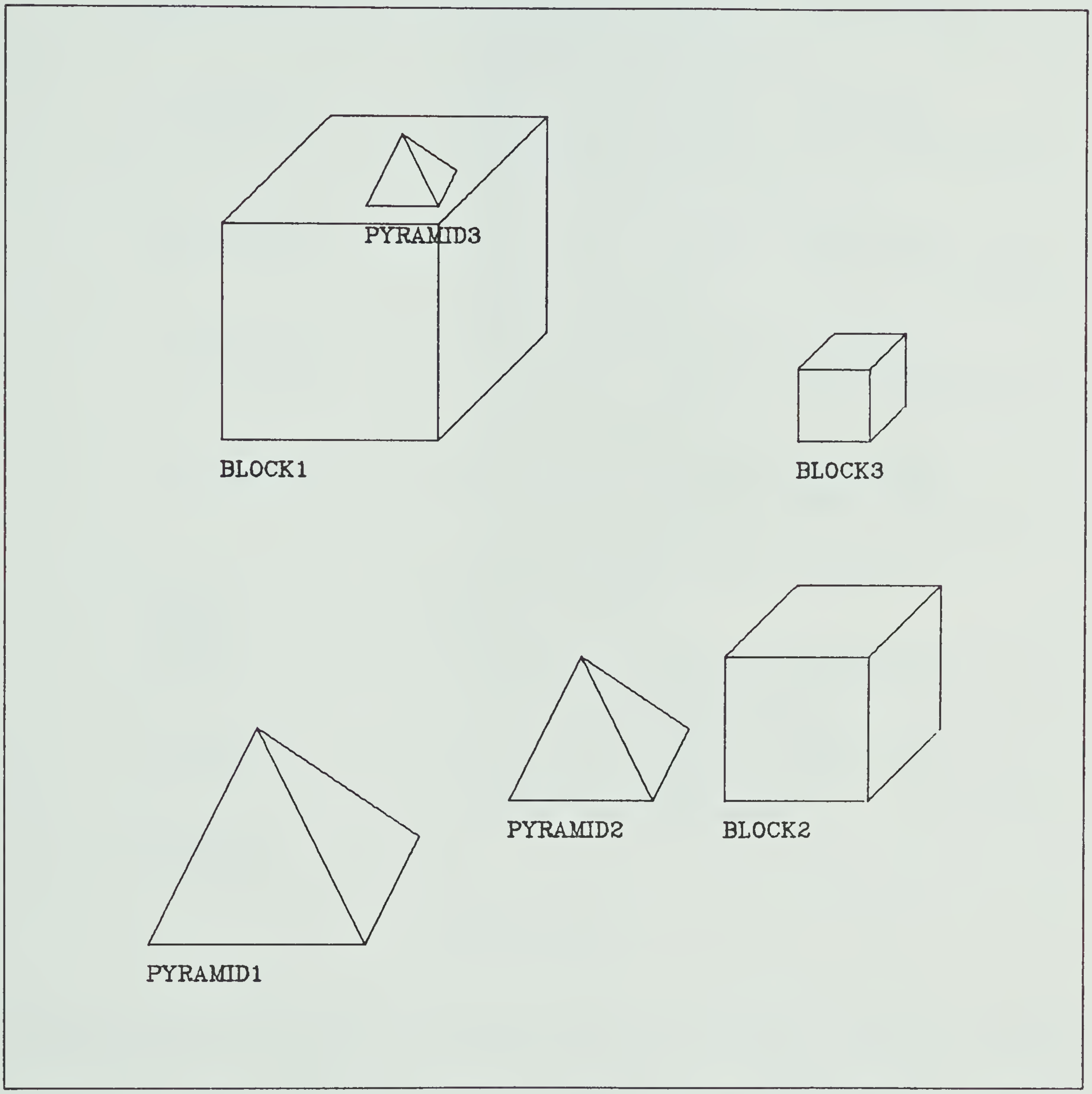


Figure 4.1

be resting on a table. The distinguishing characteristics of the objects are those of size, color and shape. The

```
                    Initial Word Knowledge

    (DEFPROP GRAMMAR ((
       (BLOCK1           ((FUNC OBJ) (OBJECT PHYSICAL)
                          (LOCATION UNSPEC) (STATE MOVEABLE)
                          (SHAPE RECTANGULAR)
                          (SIZE LARGE) (COLOR PINK)))
       (BLOCK2           ((FUNC OBJ) (OBJECT PHYSICAL)
                          (LOCATION UNSPEC) (STATE MOVEABLE)
                          (SHAPE RECTANGULAR)
                          (SIZE SMALL) (COLOR BLUE)))
       (BLOCK3           ((FUNC OBJ) (OBJECT PHYSICAL)
                          (LOCATION UNSPEC) (STATE MOVEABLE)
                          (SHAPE RECTANGULAR)
                          (SIZE TINY) (COLOR RED)))
       (PYRAMID1         ((FUNC OBJ) (OBJECT PHYSICAL)
                          (LOCATION UNSPEC) (STATE MOVEABLE)
                          (SHAPE POINTED)
                          (SIZE LARGE) (COLOR PINK)))
       (PYRAMID2         ((FUNC OBJ) (OBJECT PHYSICAL)
                          (LOCATION UNSPEC) (STATE MOVEABLE)
                          (SHAPE POINTED)
                          (SIZE SMALL) (COLOR YELLOW)))
       (PYRAMID3         ((FUNC OBJ) (OBJECT PHYSICAL)
                          (LOCATION UNSPEC) (STATE MOVEABLE)
                          (SHAPE POINTED)
                          (SIZE TINY) (COLOR BLACK)))
       (IS               ((FUNC ACT)))
       (ARE              ((FUNC ACT)))
       (TOP              ((FUNC OBJ) (OBJECT PHYSICAL)
                          (LOCATION UNSPEC) (STATE IMMOBILE)
                          (SHAPE RECTANGULAR)))
       (TABLE            ((FUNC OBJ) (OBJECT PHYSICAL)
                          (LOCATION UNSPEC) (STATE IMMOBILE)
                          (SHAPE RECTANGULAR)))
    )) VALUE)
```

Figure 4.2

accompanying initial word and world knowledge assumed by the model is given in figures 4.2 and 4.3 respectively. Sample input for this experiment can be found in figure 4.4.

The evaluated associations resulting from the first iteration with the input is summarised in table 4.1. Only

```
World Knowledge

(DEFPROP RELATT ((
(BLOCK1    ((ON TABLE) (BEHIND PYRAMID1)
            (SUPPORTS PYRAMID3)))
(BLOCK2    ((ON TABLE) (NEAR PYRAMID2) (BESIDE PYRAMID2)
            (INFRONTOF BLOCK3)))
(BLOCK3    ((ON TABLE) (BEHIND PYRAMID2) (NEAR BLOCK2)
            (BEHIND BLOCK2)))
(PYRAMID1 ((ON TABLE) (INFRONTOF BLOCK1)
            (INFRONTOF PYRAMID3)))
(PYRAMID2 ((ON TABLE)))
(PYRAMID3 ((ON BLOCK1) (BEHIND PYRAMID1)))
))VALUE)
```

Figure 4.3

```
Sample Input

(THERE IS A LARGE PINK BLOCK1 ON THE TABLE)
(THE PINK BLOCK1 WHICH IS LARGE SUPPORTS A PYRAMID3)
(A PINK BLOCK1 IS LARGE)
(THE BLOCK1 AND PYRAMID1 ARE PINK)
(THE LARGE PYRAMID1 IS INFRONTOF THE PINK BLOCK1)
(ON THE TABLE IS A LARGE PINK PYRAMID1)
(A PYRAMID1 IS INFRONTOF THE BLACK PYRAMID3)
(BESIDE THE PYRAMID2 IS A BLOCK2)
(THE PYRAMID2 AND THE BLOCK2 ARE SMALL)
```

Figure 4.4

the two highest rated associations are given for each unknown word. A "nil" indicates that no association was made, while ". . ." indicates that the association(s) made had a very low evaluation.

A rather simple heuristic was used to determine whether a correct association has been made. If the result of the

evaluation function was greater than "5", or if the primary (and possibly secondary) association was "significantly" larger than any other association, then the corresponding concepts were assumed to be the given unknown word's meaning. The value of "5" was arbitrarily chosen for experimental purposes and has no special significance. It is unknown what an optimal value would be.

First Round of Associations

unknown	primary association		secondary association	
there	nil		nil	
a	(ref indef)	17.2	(num one)	14.6
large	(ref indef)	5.0	(size large)	3.2
pink	(color pink)	5.0	(size large)	5.0
on	(physrel on)	6.5	(ref def)	1.5
the	(ref def)	22.7	(num one)	20.4
behind	(physrel behind)	2.6	(physrel infrontof)	2.6
which	nil		nil	
supports	. . .		. . .	
and	(cnj cnj)	9.1	(physrel beside)	3.5
infrontof	(physrel infrontof)	2.6	(physrel behind)	2.6
black	(color black)	1.1	. . .	
yellow	(color yellow)	2.0	. . .	
small	(ref indef)	2.9	(color yellow)	2.8
beside	(physrel beside)	1.2	(color yellow)	1.1
near	(physrel near)	2.5	(color yellow)	2.5
blue	(color blue)	1.5	. . .	
tiny	(size tiny)	6.1	(color red)	6.0
red	(color red)	3.1	(physrel near)	1.0
of	. . .		. . .	

Table 4.1

Using the above criterion, one can see from the first round of associations that the following words have been considered learned.

```
(a      ((ref indef)(num one)))
(on     ((physrel on)))
(the    ((ref def)(num one)))
(and    ((cnj cnj)))
(tiny   ((size tiny)(color red)))
```

A slight error is made in regards to the word "tiny" since the concept "(color red)" has been included as part of its meaning. The reason for this occuring is that the only red object in the given blocks world also happens to be tiny. Because of this, the model can make no further discrimination on the possible meaning of "tiny", at least with the particular environment used in the experiment.

The words considered learned in the first round of associations were then added to the model's vocabulary and another run with the given input was made. The results of the second round of associations is shown in table 4.2.

Using the same criteria as before, "pink" and "small" could be considered learned. The meaning for "pink" is only partially correct. However, since the only "pink" objects in the blocks world are also "large", it is not unreasonable for the model to conclude that "(size large)" is part of the meaning of pink. Also, for the most part, the unknown words show a positive trend towards acquiring their correct meaning.

To see if further improvement could be made with the same input and blocks world situation a third iteration was made. As before, the newly learned words were incorporated

Second Round of Associations

unknown	primary association		secondary association	
there	nil		nil	
large	(ref indef)	3.8	(size large)	3.6
			(color pink)	3.6
pink	(color pink)	5.4	(size large)	5.4
behind	(physrel behind)	3.9	(physrel infrontof)	3.9
which	nil		nil	
supports	. . .		. . .	
infrontof	(physrel infrontof)	3.9	(physrel behind)	3.9
black	(color black)	2.5	(ref def)	1.1
yellow	(color yellow)	2.9	(size small)	2.6
small	(size small)	5.3	(color yellow)	3.3
			(color blue)	3.3
beside	(color yellow)	2.3	(physrel beside)	1.7
near	(size small)	4.6	(physrel near)	3.5
blue	(color blue)	2.9	(size small)	1.8
red	(color red)	4.3	(size tiny)	2.9
of	. . .		. . .	

Table 4.2

into the model's vocabulary. The results are shown in table 4.3.

The only word that was considered learned in this case was "red", though if a decision had to be made on the others, the model would have the correct meanings for six additional words. Altogether, of the 20 words the model attempted to acquire, 13 could be considered to have been learned correctly.

It should be noted that the model has difficulty with locative concepts, in particular "behind" and "infront". This difficulty arises since, whenever we talk about an object being "behind" another, the concept "infront" is also implicitly present. To deal with this problem, the model

Third Round of Associations

unknown	primary association		secondary association	
there	nil		nil	
large	(ref indef)	3.9	(num one)	3.6
behind	(physrel behind)	4.3	(physrel infrontof)	4.3
which	nil		nil	
supports	. . .		. . .	
infrontof	(physrel infrontof)	4.3	(physrel behind)	4.3
black	(color black)	2.9	(size tiny)	2.3
yellow	(color yellow)	3.2	(size small)	2.6
beside	(color yellow)	2.7	(physrel beside)	1.8
near	(size small)	5.9	(physrel near)	3.6
blue	(color blue)	3.2	(size small)	2.6
red	(color red)	5.2	(size tiny)	2.9
of	. . .		. . .	

Table 4.3

will need to incorporate some additional discrimination strategies. Clark (1975) has made some indication as to what these strategies might be, in the context of child acquisition of prepositions. She found, to summarise briefly, that if an object is a container, then the relation of another object to the container is chosen by the child to be "in". Similarly, if an object has a supporting surface, the chosen concept would then be "on". As to the detailed formation of such strategies, in the context of the current model, much remains to be done.

Because of the size of the network produced, only the "edited" version is displayed below in figure 4.5. The original consisted of 43 nodes and 62 arcs while the network in figure 4.5 consists of 18 nodes and 20 arcs. The corresponding descriptions of the generalizations in the

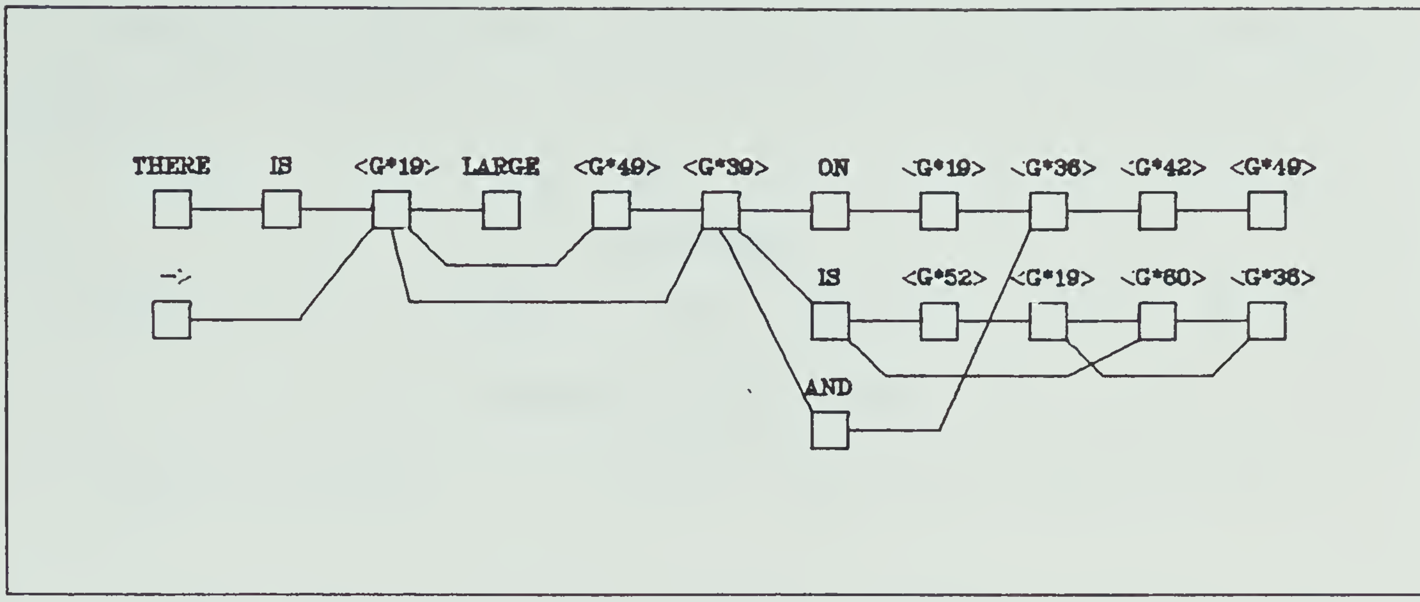


Figure 4.5

network follows in figure 4.6.

```
                    Network Generalizations

      (<G*>0019          ((FUNC REF)(REF ?)(NUM ONE)))
      (<G*>0036          ((FUNC OBJ)(OBJECT PHYSICAL)
                          (LOCATION UNSPECIFIED)(STATE ?)
                          (SHAPE ?)))
      (<G*>0039          ((FUNC OBJ)(OBJECT PHYSICAL)
                          (LOCATION UNSPECIFIED)
                          (STATE MOVEABLE) (SHAPE ?)
                          (SIZE LARGE)(COLOR PINK)))
      (<G*>0042          ((FUNC ACT)))
      (<G*>0049          ((FUNC ADJ)(SIZE ?)))
      (<G*>0052          ((FUNC PREP)(PHYSREL ?)))
      (<G*>0060          ((FUNC ADJ)))
```

Figure 4.6

The network in figure 4.5 does not represent the complete structure of all the sentences given in the input. Certain information has become lost due to the editing process. However, a significant part of possible sentence

structure has been retained as can be seen in the examples below.

sentence

There is a large pink block1 on the table

network equivalent

There is <G*19> . . . <G*49> <G*39> on <G*19> <G*36>

Every element of the sentence, except "large", has a corresponding component, possibly generalized, in the network. "Large" has to be omitted since it has no links to other parts of the network. As another example, consider

sentence

The large pyramid1 is infrontof the pink block1

network equivalent

<G*19> <G*49> <G*39> is <G*52> <G*19> <G*60> <G*36>

In this case all elements are matched, though the word "large" must take the generalized node "<G*49>" rather than the literal node "large". An example of a sentence that is not completely represented would be,

sentence

Beside the pyramid2 is a block2

network equivalent

? <G*19> <G*39> is ? ?

There is no starting node corresponding to "Beside" and for the particular instance of "is" there is no link to the

equivalents of "a" or "block2". As far as the model is concerned, its meaning of this sentence would be "the pyramid2 is". The reason for the difficulty is that the given sentence structure is uncommon in regards to the input the model has sampled. If however, the sentence had been rewritten as, "The block2 is beside the pyramid2", the model would not have experienced any such difficulty.

4.2 Experiment 2

Experiment 2, in contrast to experiment 1, did not make use of any initial generalizations. All of the other components of experiment 1 however, remained the same. The expected result was that experiment 2 would show a slower rate of meaning acquisition as well as a greatly larger final network. The results of the first round of associations produced in experiment 2 are shown in table 4.4. Surprisingly enough, the same number of words were acquired as in the first experiment though some of the words were different. Of the words considered learned, only "large" and "pink" were slightly in error. In both cases, the concepts "(color pink)" and "(size large)" were thought to be the corresponding meanings. This inability to distinguish the correct meaning is due to the sample blocks world; the only large objects happen to be pink.

As before, a second round of associations was initiated, with the results shown in table 4.5. In this

First Round of Associations

unknown	primary association		secondary association	
there	nil		nil	
a	(ref indef)	17.2	(num one)	14.6
large	(size large)	6.2	(color pink)	6.2
pink	(size large)	9.3	(color pink)	9.3
on	(physrel on)	6.5	(ref def)	1.5
the	(ref def)	22.7	(num one)	20.4
behind	(physrel behind)	2.6	(physrel infrontof)	2.6
which	nil		nil	
supports	. . .		. . .	
and	(cnj cnj)	9.1	(physrel beside)	3.5
infrontof	(physrel infrontof)	2.6	(physrel behind)	2.6
black	(color black)	1.1	(size tiny)	1.1
yellow	nil		. . .	
small	(ref indef)	2.9	(physrel beside)	1.7
beside	(physrel beside)	1.2	. . .	
near	(physrel near)	2.5	(physrel beside)	1.6
blue	. . .		. . .	
tiny	(size tiny)	3.0	(color black)	3.0
red	(physrel near)	1.0	. . .	
of	. . .		. . .	

Table 4.4

case, four other words were considered acquired as to only two in experiment 1. To see if this trend was to continue a third round of associations was made. However this time no new words were acquired and apparently further iterations would result in similar findings, (see table 4.6). If a decision had to be made on the remaining words, only two would have been correct. Altogether, of the 20 words the model attempted to acquire, 12 could be considered to have been learned correctly as opposed to 13 in experiment 1. Interestingly enough, of the words not acquired in experiment 1, none were descriptions of objects while for experiment 2, four were. It appears then that the overall

Second Round of Associations

unknown	primary association		secondary association	
there	nil		nil	
behind	(physrel behind)	4.3	(physrel infrontof)	4.3
which	nil		nil	
supports	. . .		. . .	
infrontof	(physrel infrontof)	4.3	(physrel behind)	4.3
black	(physrel behind)	1.2	(physrel infrontof)	1.2
			(num one)	1.2
yellow	. . .		. . .	
small	(ref def)	3.4	(num one)	3.3
beside	(physrel beside)	1.7	(physrel near)	1.5
near	(physrel near)	3.5	(physrel beside)	1.7
blue	(ref def)	1.5	(num one)	1.4
tiny	(size tiny)	6.9	(color red)	6.9
red	(color red)	4.3	(size tiny)	2.9
of	. . .		. . .	

Table 4.5

Third Round of Associations

unknown	primary association		secondary association	
there	nil		nil	
which	nil		nil	
supports	. . .		. . .	
black	(num one)	1.5	(ref def)	1.5
yellow	nil		nil	
small	(num one)	3.6	(ref def)	3.6
beside	(physrel beside)	1.7	(physrel near)	1.5
near	(physrel near)	3.6	(physrel beside)	1.8
blue	(num one)	1.6	(ref def)	1.6
of	. . .		. . .	

Table 4.6

effect of concept generalization on word acquisition is crucial only for descriptive words. The effect on the growth of the network is slight. Before editing, the network of experiment 2 consisted of 44 nodes and 69 arcs as

opposed to 43 and 62 in experiment 1. The edited network of experiment 2 is shown in figure 4.7.

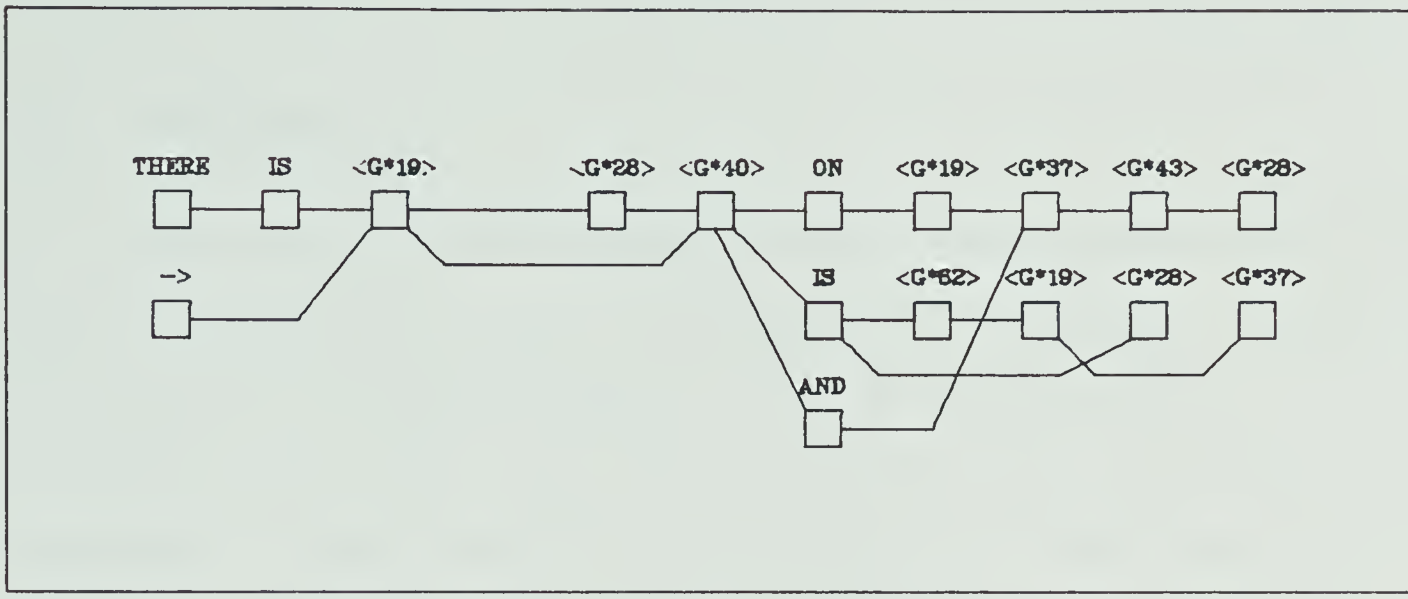


Figure 4.7

The corresponding generalizations for network 2 are given in figure 4.8.

```
                    Network Generalizations

    (<G*>0019        ((FUNC DET)(REF ?)(NUM ONE)))
    (<G*>0028        ((FUNC ADJ)(SIZE LARGE)
                      (COLOR PINK)))
    (<G*>0037        ((FUNC OBJ)(OBJECT PHYSICAL)
                      (LOCATION UNSPEC)(STATE ?)
                      (SHAPE ?)))
    (<G*>0040        ((FUNC OBJ)(OBJECT PHYSICAL)
                      (LOCATION UNSPEC)(STATE MOVEABLE)
                      (SHAPE ?)(SIZE LARGE)(COLOR PINK)))
    (<G*>0043        ((FUNC ACT)))
    (<G*>0062        ((FUNC PREP)(PHYSREL ?)))
```

Figure 4.8

The main difference in the two networks is one of more

complete generalization in network 1. As far as the model is concerned, the first network is more complete.

4.3 Experiment 3

Experiment 3 was designed to test the stability of the model's performance with a different set of input. It was expected that different words would be learned in a different order than that of experiment 1 and that the resulting network would also be different. What was not known, was whether a similar number of words would be acquired, or whether the corresponding network would contain as much information as the one in experiment 1.

The blocks world situation used in experiment 3 is shown in figure 4.9. The initial grammar, world knowledge and input is similar to that of experiment 1 and so is not shown here.

The results of the first round of associations is shown in table 4.7. Here, according to the model's heuristics, seven words were considered learned: "a", "pink", "and", "the", "tiny", "large" and "beside". This is a slightly larger total than those obtained in the first two experiments, but not significantly so. Continuing as before, a second round of associations was made. The results are in table 4.8.

In this case four additional words were acquired:

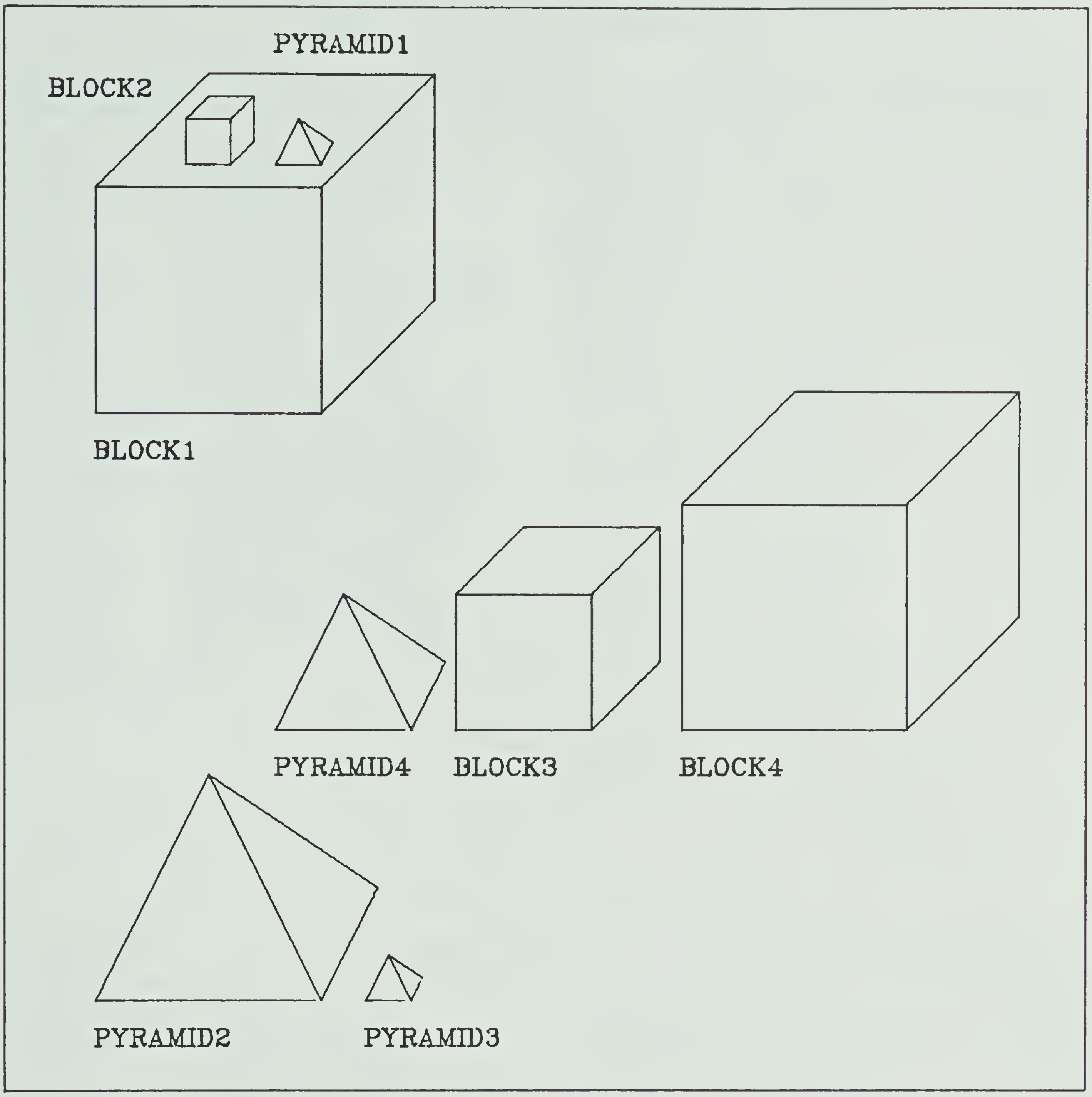


Figure 4.9

"supports", "on", "yellow" and "red". "Yellow" and "on" were considered learned despite relatively low association values, because there were no other associations made. A major error was made in regards to "red" as the model thought that its meaning was either the relation "beside" or "near". The reason for this error is that the red block in the given situation happened to be "beside" and "near" two

First Round of Associations

unknown	primary association		secondary association	
a	(ref indef)	20.1	(num one)	12.7
pink	(color pink)	8.8	(size large)	6.4
supports	(physrel on)	4.0	(ref indef)	1.6
and	(cnj cnj)	10.3	(color yellow)	4.4
the	(ref def)	34.3	(num one)	31.3
behind	(physrel behind)	2.6	(color red)	1.2
on	(physrel on)	2.6	(color pink)	1.4
blue	(color blue)	2.8	(physrel near)	1.7
tiny	(size tiny)	8.9	(color yellow)	7.7
infrontof	(physrel behind)	1.3	(size small)	1.2
large	(size large)	5.3	(color blue)	5.1
beside	(physrel beside)	5.1	(color blue)	1.6
yellow	(color yellow)	1.2	nil	
red	(physrel near)	3.5	(color red)	3.1
small	(size small)	3.1	(color black)	3.1
black	(color black)	0.8	(size small)	0.8
near	(physrel near)	3.2	(color red)	1.2
there	nil		nil	

Table 4.7

Second Round of Associations

unknown	primary association		secondary association	
supports	(physrel on)	5.2	(size tiny)	3.6
behind	(size large)	3.7	(physrel behind)	3.6
on	(physrel on)	3.5	nil	
blue	(color blue)	3.6	(size tiny)	3.6
infrontof	(physrel behind)	1.8	(size small)	1.8
yellow	(color yellow)	1.8	nil	
red	(physrel beside)	5.2	(physrel near)	3.7
small	(color black)	3.7	(num one)	1.9
black	(size small)	1.7	(color black)	1.6
near	(physrel beside)	4.7	(physrel near)	3.4
there	nil		nil	

Table 4.8

other objects. Thus whenever the red block was discussed, emphasis was placed on its relational aspects with the other

objects.

As before one final round of associations was made and the results are summarised in table 4.9.

Third Round of Associations

unknown	primary association		secondary association	
behind	(size large)	3.7	(color yellow)	3.7
blue	(color blue)	3.8	(size tiny)	3.8
infrontof	(physrel behind)	1.8	(size small)	1.8
small	(color black)	3.7	(num one)	2.0
black	(size small)	1.7	(color black)	1.6
near	(physrel near)	1.9	(physrel beside)	1.9
there	nil		nil	

Table 4.9

As can be seen no additional words were acquired, but if the model had to make a decision now it would be right in three of the remaining seven cases. Altogether 13 of the 18 words to be learned were acquired; a result in line with the first two experiments. More of the relational words were acquired in experiment 3, but this is due to more emphasis being placed on such words in the corresponding input.

4.4 Experiment 4

The evaluation function (see page 16) has a central role in the performance of the model. However, it contains the constant "m" whose value was never fully explained by McMaster, other than to say it was determined empirically to

give the "best" results. It is known however that as m increases from 0 to 1 the values obtained become smaller as well as their absolute differences.

To determine its effect in the current model, different values of m were selected, holding everything else constant. Conditions identical to the first round of iterations for Experiment 1 were obtained, and then m was varied prior to evaluation. The results are are summarised in table 4.10.

Word	Primary Concept	Values of m .0525	.105	.21	.42	.84
There	nil	nil•	nil•	nil•	nil•	nil•
A	ref indef	20.1+	18.6+	17.2+	14.5+	8.9+
Large	ref indef	7.3•	6.5•	5.0•	2.0+*	nil•
Pink	color pink	5.7•	5.4•*	5.0+	4.0+	1.9+
On	physrel on	7.6+	7.2+	6.5+	4.9+	1.8+
The	ref def	25.2+	24.4+	22.7+	19.4+	12.9+
Behind	physrel behind	5.2+	4.3+	2.6+	nil•	nil•
Which	nil	nil•	nil•	nil•	nil•	nil•
Sup'rts	nil	nil•	nil•	nil•	nil•	nil•
And	cnj cnj	9.6+	9.2+	8.4+	6.7+	3.4+
Infront	physrel infront	5.2+	4.3+	2.6+	nil•	nil•
Black	color black	3.3+	2.5+	1.1+	nil•	nil•
Yellow	color yellow	3.5+	3.0+	2.0+	nil•	nil•
Small	cnj cnj	3.8+*	3.5•*	3.1•	2.2•	nil•
Beside	physrel beside	1.8•*	1.6•*	1.2+	nil•	nil•
Near	physrel beside	1.9•*	1.8+*	1.6•	1.2•	nil•
blue	color blue	3.4+	2.7+	1.5+	nil•	nil•
Tiny	size tiny	10.5+	9.1+	6.1+	nil•*	nil•
Red	color red	5.3+	4.5+	3.1+	nil•*	nil•
Of	nil	nil•	nil•	nil•	nil•	nil•

Table 4.10

In the table the Primary Concept listed corresponds to the one selected with m equal to .21. A "+" indicates that

the correct concept has been chosen; a "•" is used for an incorrect concept. Thus for m equal to .21, there are 13 correctly chosen concepts versus 7 incorrect ones. A "*" is used to indicate a change in the concept selected, though this new concept is not shown. For example, "large" is initially associated with "reference indefinite" which is incorrect. By doubling the value of m, the evaluation dropped from 5.0 to 2.0 and some other concept was found to have a higher evaluation. Since this other concept was the desired one, a "+" and a "*" are found in the .42 column. Similarly, "beside" is initially associated with "physrel beside" which is correct. By halving the value of m, the evaluation increased from 1.2 to 1.6, but some other incorrect concept emerged with a higher rating. In this case, a "•" and "*" are used to indicate this change. The totals for the five values of m are listed below in table 4.11.

	.0525	.105	.21	.42	.84
no. correct	12+	12+	13+	6+	5+
no. incorrect	8•	8•	7•	14•	15•

Table 4.11

As can be seen from table 4.11, choosing a value of .21 for m, resulted in the highest number of correct concepts being selected. It is not unreasonable to assume though,

that as the model acquires additional word meanings, that a different value of m might lead to better results. The above test used data only from the preliminary round of associations and hence is not conclusive.

4.5 Experiment 5

The scope of the previous experiments was somewhat restrictive in that only a simple and well-defined blocks world was dealt with. Often it is advantageous to use such a paradigm so as to gain a clear understanding of what the individual components of a model add to the whole, without attempting to determine the side effects of the experimental medium. However, for a model to have any significance, it must be shown eventually that it is also able to maintain its explanatory power in worlds not quite so artificial. The following experiment endeavours to deal with just such a situation.

The world knowledge for the experiment consists of information relating to several species of animals such as bears, cats, turtles and owls. In processing data on this animal world the model attempts to acquire sets of concepts which relate common characteristics shared between different animals. In addition, the model tries to acquire those words which describe these characteristics, i.e. mammal, bird, pet, etc. A typical entry in the initial world knowledge is considerably more detailed than any in the

blocks world as can be seen by the definition for a "bear" below,

```
(bear ((func animateobject) (covering fur)
     (eathabits herbivorous) (activetime diurnal)
     (environment terrestial) (relationtoman nonpet)
     (movement walks) (sounds growls) (face nose)
     (foot claws) (bodytemperature warmblooded)
     (fur brown) (herbivorous greens)
     (diurnal unspecified) (terrestial north)
     (nonpet dangerous) (walks quadraped) (growls loud)
     (nose pointed) (claws nonretractile)
     (warmblooded unspecified)))
```

```
THE WOLF AND THE OPOSUM ARE MAMMAL
THE PORCUPINE IS MAMMAL AND THE VULTURE THE BIRD
A BEAR IS A NONPET
THE CAT IS THE PET AND A DOG IS A PET
THE OWL IS THE BIRD
A CAT IS A MAMMAL
A DOG IS A MAMMAL
A CROCODILE IS A REPTILE
THE BEAR IS THE MAMMAL
A OPOSUM IS A NONPET
A TURTLE AND A CROCODILE ARE REPTILE
A TURTLE IS A REPTILE
THE CROCODILE IS THE NONPET
A VULTURE IS A NONPET
THE TURTLE AND THE CROCODILE ARE REPTILE
```

Figure 4.10

Since the concept attributes "covering" through to "bodytemperature" would more or less be common for all animals they were not used in the experiment. Similar entries to the above were made for the following animals: opossum, dog, wolf, porcupine, cat, turtle, crocodile, owl and vulture. The input sentences presented to the model are

shown in figure 4.10.

The sets of concepts and the order in which they were generated can be found in figure 4.11. One can see that the

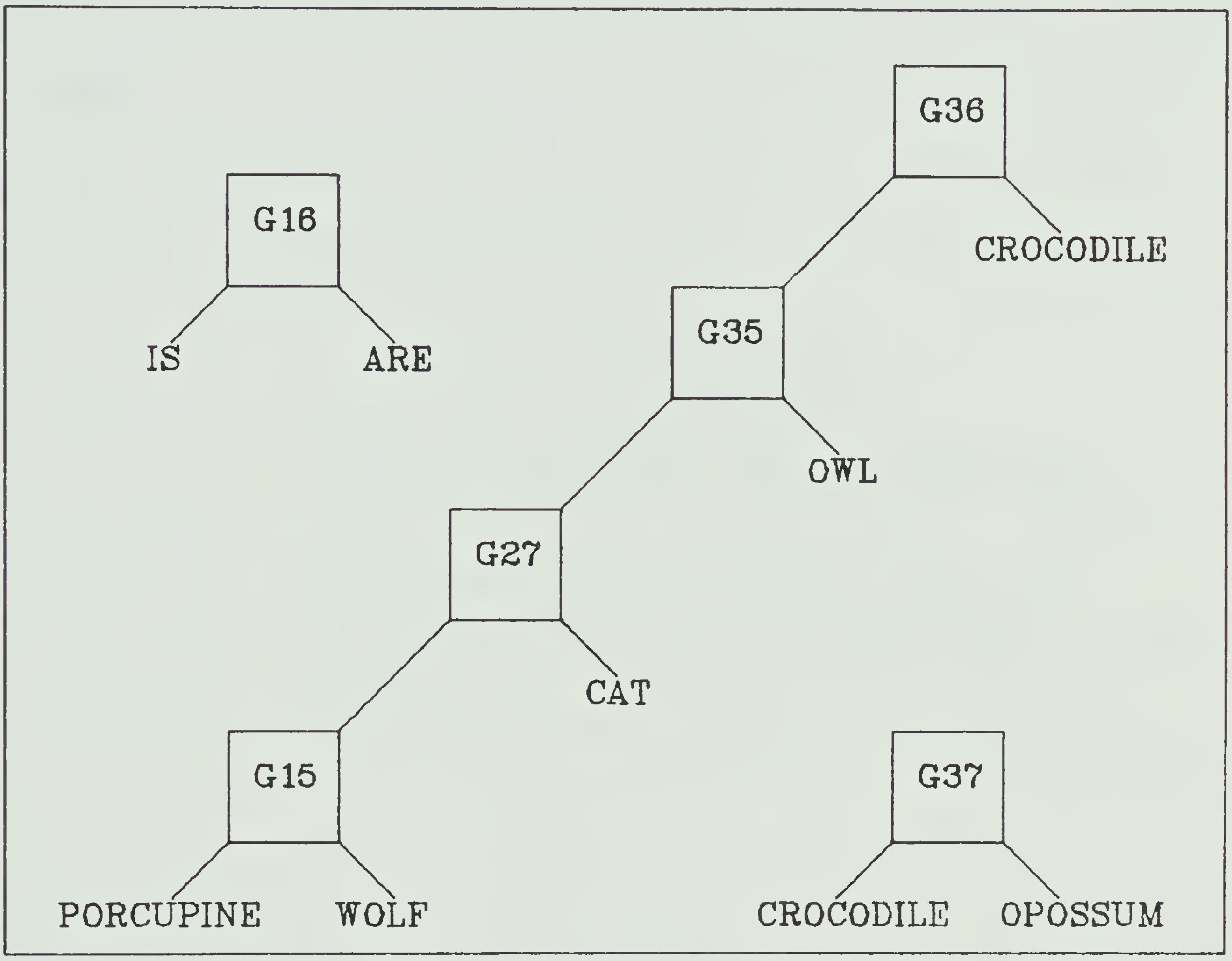


Figure 4.11

concepts associated with a porcupine and a wolf were used to generate a new set of concepts (G15) which was later used, along with the concepts associated with a cat, to generate the set "G27". The most generalized set formed (G36) is a simple indication that something is an animate object. The composition of these generated sets of concepts can be found

in figure 4.12. Due to the small amount of input and the relative similarity of sentence construction not all possibly useful generalizations were formed. That is, since there were a wide variety of very different animals the generalization process quickly accelerated to the very general concept "G36", thus bypassing some possibly useful intermediate generalizations. More will be said below on how these generalizations may be associated with words considered to be known by the model. The word-concept

```
                    Concept Generalizations

(<G*>0015    ((FUNC OBJ) (TERRESTIAL ?)
              (NONPET DANGEROUS) (WALKS QUADRAPED)
              (NOSE ?) (CLAWS ?) (WARMBLOODED UNSPEC)))
(<G*>0016    ((FUNC ACT)))
(<G*>0027    ((FUNC OBJ) (TERRESTIAL ?) (WALKS ?)
              (NOSE ?) (CLAWS ?) (WARMBLOODED)))
(<G*>0035    ((FUNC OBJ) (CLAWS ?) (WARMBLOODED UNSPEC)))
(<G*>0036    ((FUNC OBJ)))
(<G*>0037    ((FUNC OBJ) (CARNIVOROUS ?)(NONPET ?)))
```

Figure 4.12

associations formed are summarised in table 4.12. Using the same decision criteria as in the previous experiments the model would consider "the", "and" and "a" to be learned. Of the other words, "bird" and "reptile" seem to be closest to being properly recognized. As before, the learned words were added to the model's world knowledge and a second set of associations were formed. The results of this second round of associations are summarised in table 4.13.

From these results the only other word to have been

First Round of Associations

The	- ref def(12.5), num-one(10.8), nonpet dangerous(8.9) claws nonretractile(8.9), beak sharp(7.4), . . .
Mammal	- ref def(4.1), num one(3.1), walks quadraped(2.9), terrestial houses(2.9), . . .
And	- cnj cnj(7.3), amphibian south(3.0), scales green(2.6), snout big(2.6), . . .
Bird	- feathers soft(1.7), beak sharp(1.7), carnivorous mice(1.7), aerial barnyards(1.7), . . .
A	- ref indef(16.1), num one(14.5), mixeddiet purina(11.1) terrestial houses(11.1), . . .
Nonpet	- ref indef(3.2), walks quadraped(2.6), nose pointed(2.5), nonpet dangerous(2.4), . . .
Pet	- nose wet(2.1), walks slowly(2.1), mixeddiet purina(1.3), terrestial houses(1.3), . . .
Reptile	- amphibian south(4.1), scales green(3.3), snout big(3.3), ref indef(3.2), . . .

Table 4.12

learned would be "reptile". However, if the model were to make a decision regarding the other words it would have reasonably accurate definitions for all of them. In contrast to the blocks world, definitions for the learned words are not as simple or consise, i.e. the definition for a mammal would probably consist of the 8 concepts listed below. Also, the definitions consist of some rather general concepts ("walks quadraped") as well as some rather specific

Second Round of Associations

Mammal	- walks quadraped(3.7), nose pointed(3.7), mixeddiet purina(3.7), terrestial houses(3.7), pet friendly(3.7), fur brown(3.6), claws nonretractile(3.6), nonpet dangerous(3.6), . . .
Bird	- feathers soft(3.2), beak sharp(3.2), . . .
Nonpet	- walks quadraped(3.6), nose pointed(3.6), claws retractile(3.5), nonpet dangerous(3.5), . . .
Pet	- walks slowly(3.6) nose wet(3.6), pet friendly(3.2), terrestial houses(3.2), mixeddiet purina(3.2), . . .
Reptile	- amphibian south(5.4), scales green(5.2), snout big(5.2), croaks loud(4.6), . . .

Table 4.13

concepts ("terrestial houses") not to mention some seemingly contradictory concepts, (Mammal - pet friendly and nonpet dangerous). Actually, such a situation seems to be an accurate reflection, to some degree, as to the sort of concepts held by people.

The words the model was attempting to acquire in this experiment can be associated to some of the generalizations found in figure 4.10. The details of such a procedure have not been worked out but a possible solution could be as follows. One could take the set of concepts comprising a word's definition and then determine which generalization matches these concepts the closest without any contradictions. For example, five of the concepts

associated with "mammal" match those of "G15" but there is a contradiction between "pet" and "nonpet"; four concepts match those of "G27" without contradiction; and only one concept matches "G35". Hence it would not be unreasonable to rename "G27" to "mammal". By a similar reasoning "G15" could be renamed "nonpet" which fits in very nicely since a "nonpet" is but a subset of the more generalized set "mammal".

The results of the experiment did point out one possibly important omission in the model. As was indicated earlier, the definitions for the animals were significantly more detailed than for any objects in the blocks world, yet it is not unreasonable to assume that they could have been even more detailed. Such a situation would raise some difficulties in the model's performance efficiency and so, should be examined a little closer. One solution would be to have the model weight concepts as to their salience and to only consider those of a certain weight in the early stages of acquisition. An analogous situation can be found in child language acquisition were the concept of "height" always takes precedence over "width" which takes precedence over "depth" regardless of the object being viewed, (Clark, 1977). This seems to be a useful heuristic and may even be necessary in the early stages of acquisition.

Chapter 5

CONCLUSION

Several models of various aspects of computational language acquisition were examined and commented on. The major difficulty in evaluating such work stems from the fact that they all, the current research included, deal only with simplistic child-like language. The ommission of how such models can develop the ability to make the transition to adult-like language is a serious defect. This criticism can be mollified somewhat by the realization that their exists a vast gap between what is known of language development and what is required to model such development computationally. Nevertheless, for a model to have any significance, it should be open-ended enough and flexible enough to incorporate new knowledge as it becomes available. It is felt that Reeker's Problem Solving Theory and hopefully, the current model, fall into this category.

The intent of the current research was to devise and test a computational model of language acquisition, which would have a greater flexibility and independence of operation than has been shown in any other model. In addition, a demonstration of how some of the sub-tasks of language acquisition interact with the overall acquisition process was presented. It was felt that the model was

partially successful in that it was able to attach correct meanings to words without direct referents in an environment, induce a rudimentary grammar, and to a limited extent delve into the question of cognitive development. In particular, experiments were constructed to demonstrate the model's general performance, examine the effect of generalization, test the model's stabilty to varying input, explore the influence of the evaluation function, and to determine the kind of conceptual development that the model could handle. However, it is only too evident that much remains to be done.

One of the more immediate problems with the model were the number of unresolved issues surrounding the editing of the network. Earlier it had been stated that it was not known at what times the editing process should be invoked. Three plausible alternatives would be: 1) on a periodic basis; 2) at a threshold point; and 3) not at all.

Editing on a periodic basis could occur at some natural point such as an extended break in the input. However, since the process of editing involves an examination of the entire network too short a period would be inefficient. It is even doubtful that an optimal period could be obtained since the structure of the network is dependent more on the nature of the input rather than on the age of the network itself.

A threshold point, corresponding to the available space

set aside for the network (memory size) is a somewhat appealing alternative. When memory becomes "full" the model would then envoke a compaction scheme to reclaim needed space. The one danger with this approach would be in the allowance of "too" large a memory size; a situation which could adversely affect the other processes of the model by extending their memory search times. Perhaps a measure of compaction frequency would indicate whether a given memory size was suitable or not.

The final suggested alternative was to never bother with editing at all, implying that there will always be enough memory to hold the network. This approach is appealing in that there is nothing to implement, but it is really the worst case of too large a memory size.

Once it has been decided that editing is a useful procedure, then suitable cut-off criteria for usage and age must be determined. One possibility is to simply use arbitraty values, as in the current implementation, and then to adjust them according to the performance of the model. While this method is easy to implement, it is not very theoretically useful. Alternatively, it should be possible to take average measures of the network as a whole and then use these figures to calculate which fragments of the network to remove. Such measures could include the average age of network components as well as the frequency of change in the network. If it is desirable to trim back the network

size by 5% then with the above information it would be easy to calculate the appropriate cut-off criteria. The major difficulty with this approach would be the cost of maintaining current measures of the network. Though if such measure taking were done infrequently the approach probably would be practical.

It was also suggested earlier that the removal criteria should probably change as the network ages. If the second method of determining the criteria outlined above were used then this would occur somewhat automatically. However, the percentage of the network that is to be removed should probably be less in an older network. One method of determining this would be to watch for a widening gap between the relative ages of network components, i.e., between the stable components of the network and those that are systematically removed. The occurrence of such a divergence would be one indication that the percentage of the network removed through editing should be lowered.

Whether or not the trade-off of processing efficiency for information loss through editing is profitable will depend on how accurately the above methods are implemented. The best way to determine this is to simply measure the space and time usage of the model with and without the editing procedure.

Another shortcoming of the model was the lack of consideration given to adverbials and tense. The major

difficulty the model would have in attempting to acquire such knowledge arises from the semantic representation used. Somehow a means of incorporating state changes and time markers would have to be introduced before the model could even consider acquisition of these features of language. Also there was no allowance made for the acquisition of quantifiers such as "few", "some" and "many". The difficulty in dealing with such fuzzy concepts is that the model would need a large number of samples to properly discriminate their probable meaning.

As has been mentioned earlier, there remains much work to be done in the computational modelling of language acquisition. All models still deal with only the single sentence and thus avoid problems of pronoun reference and connected discourse. There has been no attempt to handle the acquisition of the procedural aspects of language such as asking and answering questions, or the performing of commands. Many of these problems will require a much better understanding of cognitive development, and its effect on language acquisition, and other learning behavior.

REFERENCES

Anderson, J. R. and Bower, G. H. (1973). Human Associative Memory, V. H. Winston and Sons, Washington, D. C.

Anderson, J. R. (1977). "Induction of Augmented Transition Networks", in Cognitive Science 1, pp 125-157.

Bischof, W. f. (1976). "Psychology of Language and Memory", in Computational Semantics, E. Charniak and Y. Wilks (eds), North-Holland Publishing Company, pp 185-203.

Bloom, L., Rocissano, L. and Hood, L. (1976). "Adult-Child Discourse: Developmental Interacton between Information Processing and Linguistic Knowledge", in Cognitive Psychology 8, pp 521-552.

Braine, M. D. S. (1971). "On two types of models of the internalization of grammars," in The ontogenesis of grammar, D. I. Slobin (ed), Academic Press, New York.

Brown, R. (1972). A First Language, Harvard University Press, Cambridge, Massachusetts.

Charniak, E. (1976). "Inference and Knowledge Part 1", in Computational Semantics, E. Charniak and Y. Wilks (eds), North-Holland Publishing Corporation, pp 1-21.

Chomsky, N. (1965). Aspects of the theory of syntax, MIT Press, Cambridge, Massachusetts.

Clark, E. (1975). "Non-linguistic strategies and the acquisition of word meanings", in Cognition, v 2, n 2, pp 161-182.

Clark, E. (1977). "Strategies and the Mapping Problem in First Language Acquisition", in Language Learning and Though, J. MacNamara (ed), Academic Press, New York, pp 147-168.

Gardner, B. T. and Gardner, R. A. (1975). "Evidence for sentence constituents in the early utterances of child and chimpanzee", in Journal of Experimental Psychology: General, 104, pp 244-267.

Granger, R. H. (1977). "FOUL-UP: A Program that Figures Out Meanings of Words from Context", Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Boston, Massachusetts, pp 172-179.

Itagaki, (1976). "Computer Simulation of Language

Acquisition", Tokyo University of Science, Department of Industrial Administration, Japan.

Jackson, P. C. (1974). Introduction to Artifical Intelligence, Petrocelli/Charter, New York.

King, J., McMaster, I. and Sampson, J. R. (1976). "Computer Acquisition of Natural Language: Experimental Tests of a Proposed System", in COLING 76, #47.

King, J. (1976). "Correlating Verbs With Actions in a Paradigm for Computer Language Acquisition", MSc Thesis, Department of Computing Science, University of Alberta.

McMaster, I. (1975). A Proposal for Computer Acquisition of Natural Language", Technical Report TR-75-3, Department of Computing Science, University of Alberta.

McMaster, I., Sampson, J. R. and King, J. (1976). "Computer acquisition of natural language: a review and prospectus", in Int. J. Man-Machine Studies 8, pp 367-396.

Moeser, S. D. (1977). "Semantics and Miniature Artificial Languages", in Language Learning and Thought, J. MacNamara (ed), Academic Press, New York, pp 227-250.

Nelson, K. (1974). "Concept, Word, and Sentence: Interrelations in Acquisition and Development", in Psychological Review 81, pp 267-285.

Nelson, K. (1977). "The Conceptual Basis for Naming", in Language Learning and Thought, J. MacNamara (ed), Academic Press, New York, pp 117-136.

Quillian, M. R. (1969). "The Teachable Language Comprehender: A Simulation Program and Theory of Language", in Communications of the Association for Computing Machinery, v 12, n 8, pp 459-476.

Reeker, L. H. (1976). "The Computational Study of Language Acquisition", in Advances in Computers, v 15, pp 181-238.

Reeker, L. H. (1976). "The Interplay of Semantic and Surface Structure Acquisition", presented at International Psychology of Language Conference, Stirling, Scotland, June 1976.

Reeker, L. H. (1975). "Computer Simulation of Language Acquisition", presented at the Third International Symposium on Child Language, London, England, September.

Reeker, L. H. (1977). "Varieties of Learning in

Grammatical Acquisition", in Structural/Process Models of Complex Human Behavior, J. Scandura and C. Brainerd (eds), Sijthoff and Noordhoff, Alphen aan den Rijn, The Netherlands, pp 465-477.

Richards, I. A., et. al. (1961). Russian Through Pictures, Book 1, Washington Square Press, New York, New York.

Salveter, S. C. (1976). "Learning Structures To Represent Verb Meaning", Computer Sciences Department, University of Wisconsin.

Schank, R., and Abelson, R. (1978). Scripts Plans Goals and Understanding, John Wiley & Sons, New York.

Schank, R. (1973). "Identification of Conceptualizations Underlying Natural Language", in Computer Models of Thought and Language, Schank and Colby (eds), W. H. Freeman and Company, San Francisco, pp 187-249.

Schank, R. (ed) (1975). Conceptual Information Processing, North Holland.

Siklossy, L. (1972). "Natural Language Learning By Computer", in Representation and Meaning: Experiments with Information Processing systems, H. A. Simon and L. Skilossy (eds), pp 288-328.

Sinclair-de Zwart, H. (1973). "Language Acquisition and cognitive development", in Cognitve development and the acquisition of language, T. E. Moore (ed), Academic Press, New York, New York.

Winograd, T. (1972). Understanding Natural Language, Academic Press, New York, New York.

Winston, P. H. (1977). Artificial Intelligenc, Addison-Wesley Publishing Company, Reading Massachusetts.

Woods, W. A. (1970). "Transition Network Grammars for Natural Language Analysis", in Communications of the Association for Computing Machinery, v 13, n 10, pp 591-606.

Woods, W. A. (1973). "An Experimental Parsing System for Transition Network Grammars", in Natural Language Processing, R. Rustin (ed), Algorithmics Press, Incorporated, New York, New York, pp 11-154.